MCR-81-576

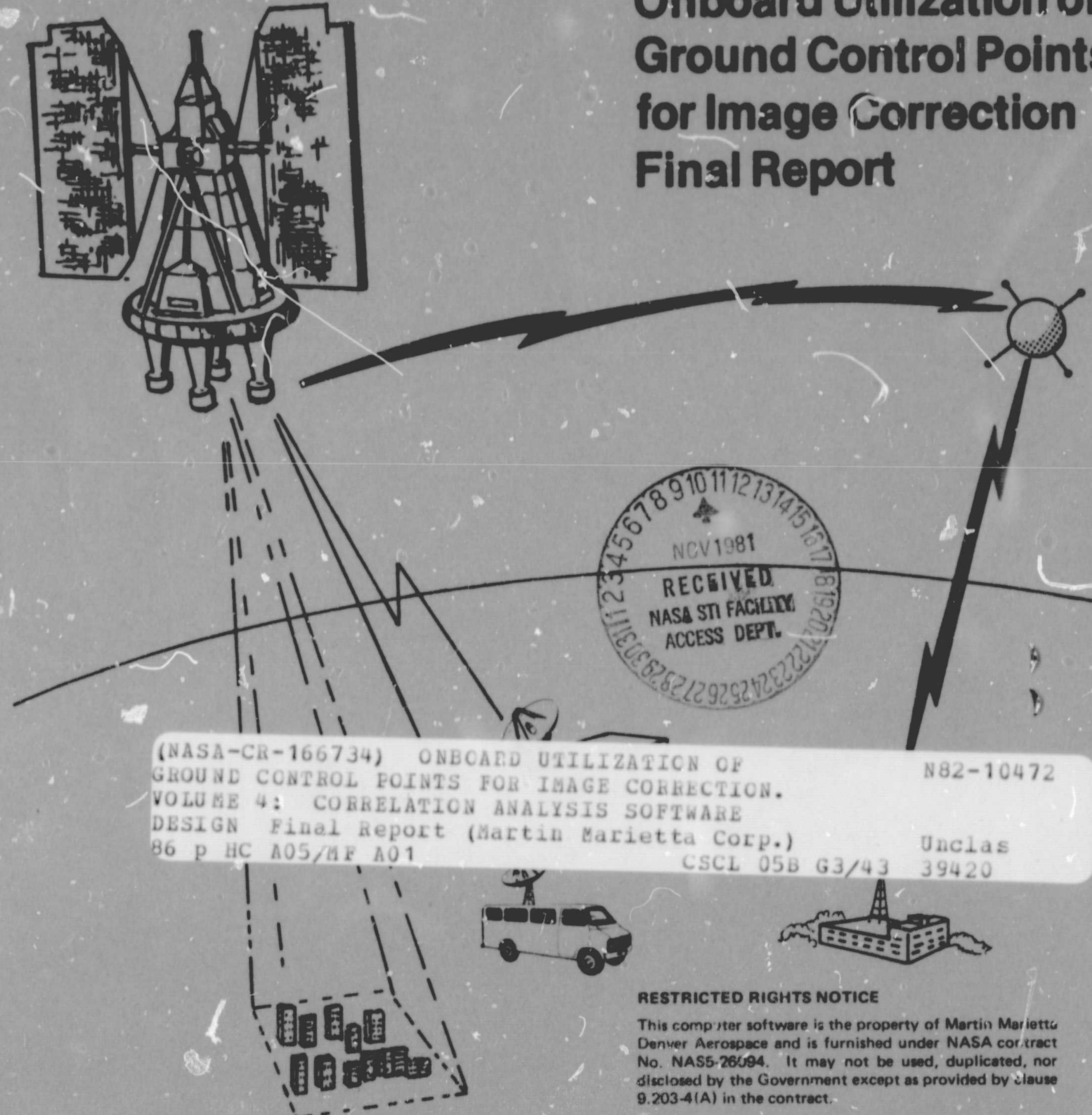
NASA CR-166734

Volume IV

Correlation
Analysis
Software Design

May 1981

Onboard Utilization of Ground Control Points for Image Correction Final Report



(NASA-CR-166734) ONBOARD UTILIZATION OF GROUND CONTROL POINTS FOR IMAGE CORRECTION. VOLUME 4: CORRELATION ANALYSIS SOFTWARE DESIGN Final Report (Martin Marietta Corp.) 86 p HC A05/MF A01 CSCL 05B
N82-10472
Unclas
G3/43 39420

MARTIN MARIETTA

Correlation
Analysis
Software Design

Volume IV

May 1981

ONBOARD UTILIZATION OF GROUND CONTROL POINTS FOR IMAGE CORRECTION FINAL REPORT

MARTIN MARIETTA
DENVER AEROSPACE
Denver, Colorado 80201

TABLE OF CONTENTS

		Page
0.0	Abbreviations	v
1.0	General Introduction	B-1
2.0	User's Guide	B-6
2.1	Introduction	B-6
2.2	Types of Runs	B-7
2.2.1	Interactive Mode, No Data Base Storage	B-7
2.2.2	Interactive Mode, Data Base Storage	B-9
2.2.3	Batch Mode, Data Base Storage	B-11
3.0	Programmer's Section	B-12
3.1	Introduction	B-12
3.2	Variable Conventions	B-12
3.3	Common Blocks	B-12
3.3.1	DATA.COM	B-13
3.3.2	DATABS.COM	B-14
3.3.3	FILES.COM	B-15
3.3.4	HEADER.COM	B-15
3.3.5	IO.COM	B-16
3.3.6	OPTS.COM	B-16
3.3.7	Common Block Reference List	B-17
3.4	CORALA Tree Structure	B-18
3.5	Program Specific Routines	B-21
3.5.1	CORALA	B-22
3.5.2	CORINI	B-23
3.5.3	CORMID	B-25
3.5.4	CORFIN	B-26
3.5.5	ADJ	B-27
3.5.6	AUTO	B-28
3.5.7	AUTOCR	B-29
3.5.8	CLOUDC	B-30
3.5.9	CLOUDS	B-31
3.5.10	CLSCOR	B-32
3.5.11	CORG	B-34
3.5.12	CORSDA	B-36
3.5.13	CORSMN	B-37
3.5.14	CRALGS	B-38
3.5.15	CRCORR	B-39
3.5.16	DISPLE	B-40
3.5.17	DISPLY	B-41
3.5.18	DIVID	B-42
3.5.19	GRID	B-44
3.5.20	GSAHIS	B-45
3.5.21	GSASTD	B-46
3.5.22	SAREA	B-47
3.5.23	SCALE	B-48
3.5.24	SPIX	B-49
3.5.25	SPLPRO	B-50
3.5.26	WEIGHT	B-51

Page

3.6	Library Routines	B-52
3.6.1	DECDE	B-53
3.6.2	DRABOX	B-54
3.6.3	GETHDR	B-55
3.6.4	IMGIT	B-56
3.6.5	LMDISP	B-57
3.6.6	MEAN	B-58
3.6.7	NXDREC	B-59
3.6.8	POS	B-60
3.6.9	REDAT	B-61
3.6.10	REDRAW	B-62
3.6.11	SPCOEF/SPLINE	B-63
3.7	I/O Routines	B-65
3.7.1	RAMTEK 9100 I/O Routines	B-66
3.7.1.1	RESET	B-67
3.7.1.2	RMCHCK	B-68
3.7.1.3	RMERSE	B-69
3.7.1.4	RMRDIM	B-70
3.7.1.5	RMTEXT	B-71
3.7.1.6	RMWRIM	B-72
3.7.1.7	WIO	B-73
3.7.2	Joystick I/O Routines	B-74
3.7.2.1	ADJUST	B-75
3.7.2.2	INIJOY	B-76
3.7.2.3	READJY	B-77
4.0	Data Base Definition	B-78
5.0	Auxiliary Programs	B-80

0.0 ABBREVIATIONS

CORALA	Correlation Analysis
G.S.A.	Grey Scale Adjustment
IR	Infrared .85 micron
M/S	Multispectral ratio
RM9100	RAMTEK 9100
S.S.D.A.	Sequential Similarity Detection Algorithm
VR	Visual Red .65 micron

I. FOREWORD

This volume of the "Onboard Utilization of Ground Control Points for Image Correction" final report provides a detailed description of the software that was utilized for image correction accuracy measurement. This software was developed under an independent research task DIIR "Video Guidance Technology" and is included here for completeness.

Three other volumes have been incorporated into the final report. Volume I provides an executive summary, Volume II contains a detailed description of the study and simulation results, and Volume III is an appendix describing the software developed for simulation of the onboard navigation system.

1.0 INTRODUCTION

The correlation analysis program is written to allow the user various tools to analyze different correlation algorithms. These algorithms were tested using LANDSAT imagery in two different spectral bands. Image data of the same area was taken at two different times. In the imagery taken at time t, a reference landmark is extracted by the user. An attempt is then made to find the corresponding landmark in the image data taken at time t_2 by using the various correlation algorithms.

Each correlation performed returns various monitoring values. The time required for the correlation is returned as well as the correlation coefficient. Cloud percentages in both landmark and search areas are displayed. The absolute best fit element and line values are also calculated.

The user controls the flow of the CORALA programs by defining landmark sizes, search area sizes, algorithm types, and which bands to use for the correlation run. Currently, the maximum search area size is 64 x 64 pixels, and the maximum landmark size is 56 x 56 pixels. Correlation may be performed on the imagery in any of three display areas. The screen setup on the RM9100 is illustrated in Figure 1.0. In this example, the user has defined a landmark in band 1. The correlation program would attempt to find this landmark in band 1 of the data taken at time t_2.

Currently, three classification algorithms have been implemented. Table 1.0 describes two of the three algorithms. The third algorithm, S.S.D.A. with thresholding, is a modification of the S.S.D.A. algorithm. This algorithm will proceed as long as the current correlation value being computed is less than the previous minimum.

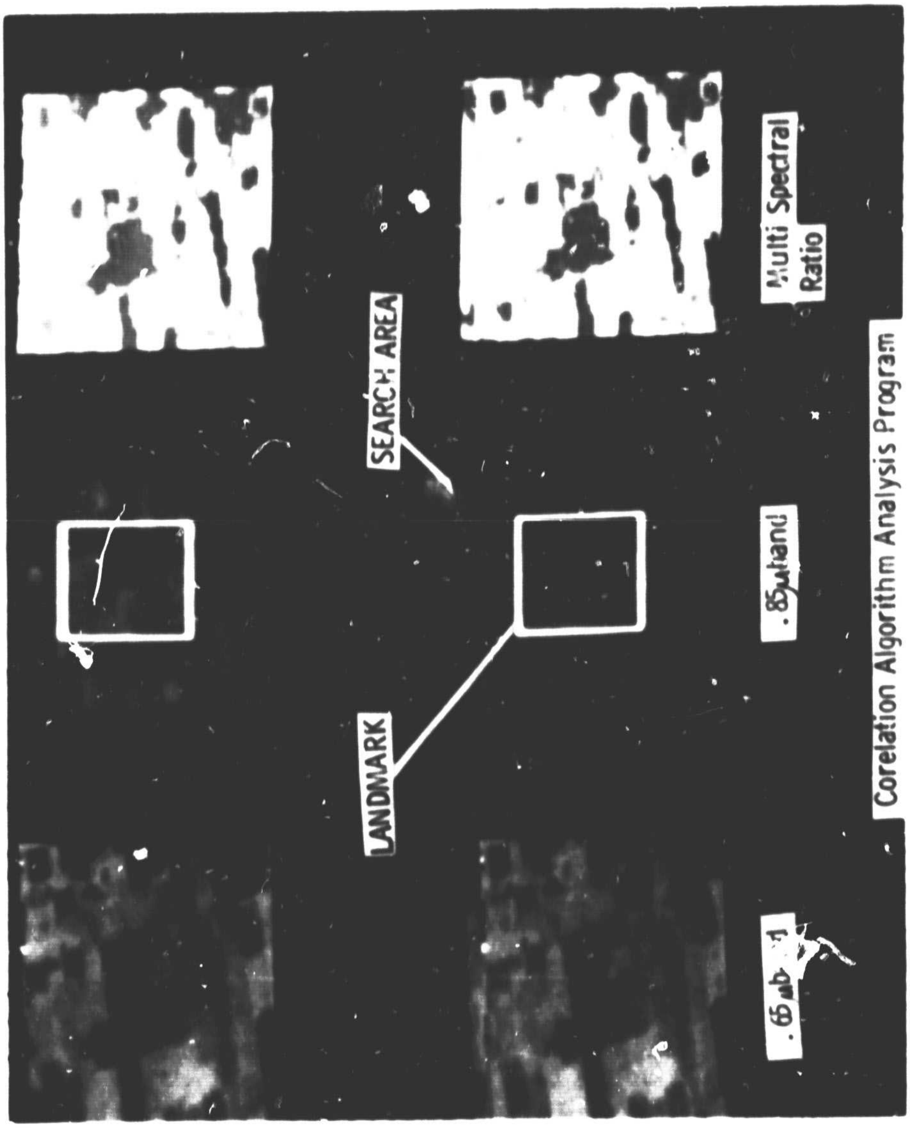


FIGURE 1.0

ORIGINAL PAGE IS
OF POOR QUALITY

$$\text{Sequential Search Detection Algorithm} = \left| \left[\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\text{Chip}(i,j) - \text{Chip mean}\right) - \left(\text{Window}(i,j) - \text{Window mean}\right) \right] \right|$$

$$\text{Classical Correlator} = \frac{\left[\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\text{Chip}(i,j) - \text{Chip mean}\right)\left(\text{Window}(i,j) - \text{Window mean}\right) \right]}{\left[\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\text{Chip}(i,j) - \text{Chip mean}\right)^2 \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\text{Window}(i,j) - \text{Window mean}\right)^2 \right]^{1/2}}$$

where

Chip(i,j) is the ijth pixel value in the n x n array of pixels representing the GCP,

Window(i,j) is the ijth pixel in the n x n segment of the m x m array of pixels representing the overlapping area of the search area and the chip position.

$$\text{Chip mean} = \sum_{i=1}^{n} \sum_{j=1}^{n} \text{Chip}(i,j)/n^2$$

$$\text{Window mean} = \sum_{i=1}^{n} \sum_{j=1}^{n} \text{Window}(i,j)/n^2$$

TABLE 1.0

CORRELATION ALGORITHMS

The primary control variables which can be altered for each particular run by the user include:

- Scene location
- GCP location
- Search area size
- GCP size

The observables on which the analysis is performed include:

- Time of correlation
- Rate of convergence
- Percentage difference in the rate of convergence for live-scene correlation versus autocorrelation
- Probability of false lock
- False lock detection feasibility
- Ease of implementation in a hardwired system
- Location of correlator lock

The quantitative results of the analysis are provided in a hard-copy print-out, and a correlation surface plot is displayed on the video monitor (Figure 1.0.1). These data provide the basis of a tradeoff between the various algorithms.

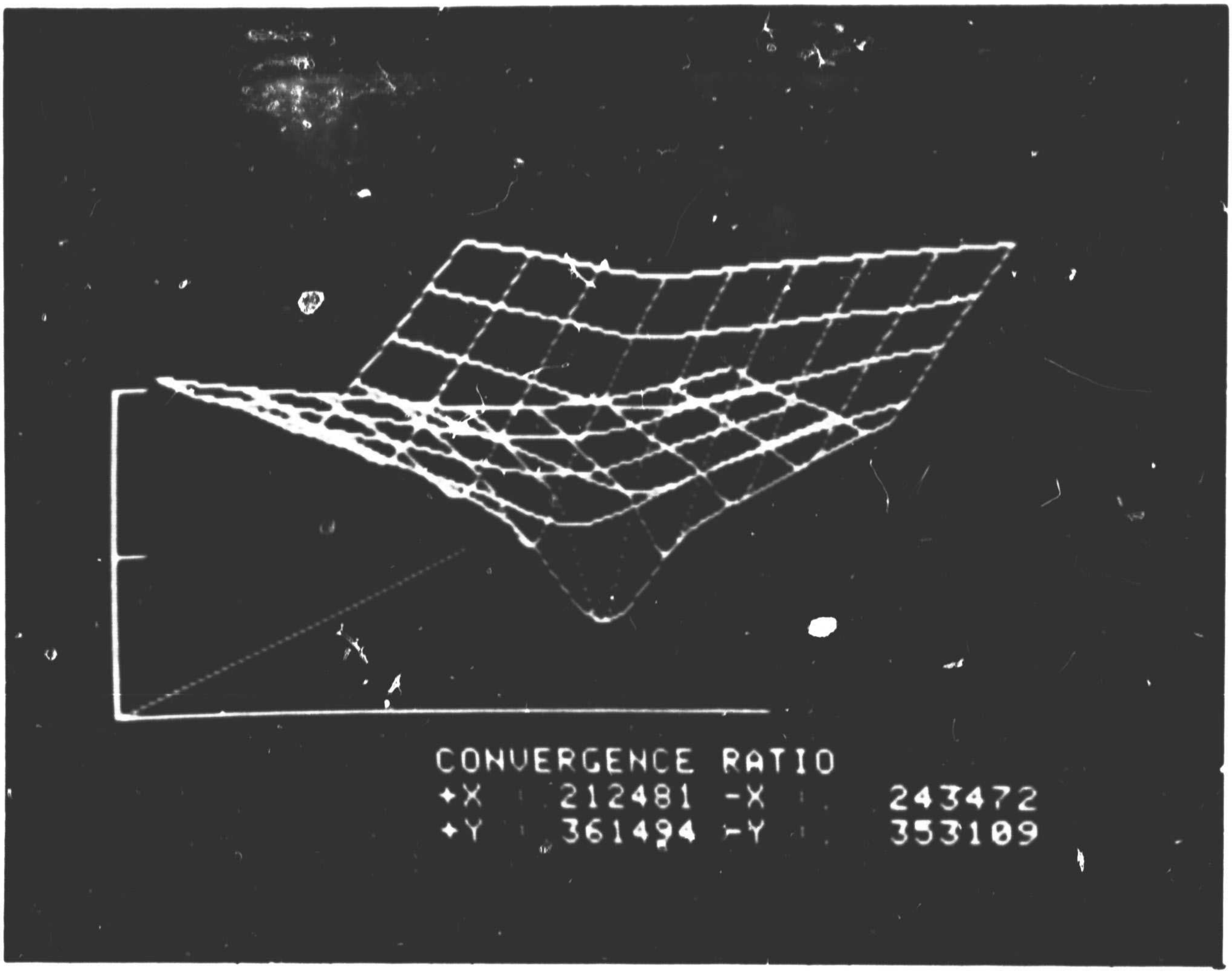


FIGURE 1.0.1

CORRELATION SURFACE PLOT

2.0 USER'S GUIDE

2.1 Introduction

The Correlation Analysis Program, CORALA, has two defined user modes; they are the interactive and the batch modes. The interactive mode queries the user for information concerning the current running of the program. Using the batch mode, this data has already been prestored, and all the user must do is initiate the running of the (program.

An auxiliary task called AUTOCOR, automatic correlation, is used by the user when batch mode is the desired mode of operation. This task basically duplicates the query section that would be required for an interactive CORALA run.

Under either mode of operation, the results of the correlation may be stored in a data base file containing previously-stored runs. The information contained in the data base is described in Section 4. When the data base storing option is selected, a predefined run sequence of landmarks, sizes, and algorithms are used.

The following types of runs may be performed:

(1) INTERACTIVE MODE, NO DATA BASE STORAGE

(2) INTERACTIVE MODE, DATA BASE STORAGE

(3) BATCH MODE, DATA BASE STORAGE

CORALA is initiated by the user by the following command sequence:

Log in to the PDP 11/70

ASN DR1: = SY:

SET/UIC = [112,5]

MOU DR1: LANDSAT

PIP *.DAT/UN/NM

RUN CORALA

From this point on, the user is prompted dependant upon the type of run being performed.

-----Double underline indicates user input.

2.2 Types of Runs

2.2.1 TYPE 1 - INTERACTIVE MODE, NO DATA BASE STORAGE

The following is the sequence of questions asked the user when under a Type 1 run:

(1) Will this run use a prestored run controller (Y or N) N

Inputting an N will cause this to be in the interactive mode.

(2) Is this run to update the corr. data base (Y or N) N

This will put the user in the no data base storage mode.

(3) Input test case number: number

This number may be ±32767.

(4) Input file name for search area 1: name B2.ext

Each image file name is designated by a name appended with a B2 or B4 signifying Band 2, .65 micron, or Band 4, .85 micron. For this question, B2 is appended. The file extension, ext, is a 3-letter file descriptor type, usually DAT.

(5) Input starting element, line, and ratio (3I5): n,n,n

Each of these numbers may be in the range ±32767. Default display ratio is 1. The element and line inputs define the absolute location of the imagery to be displayed.

(6) The user is prompted to flip switch 1 of the joystick up. Then the user positions the cursor, by using the joystick,

to the center of the desired search area. Once positioned, the user must flip switch 4 up, then down.

(7) Enter option for third display area --

1 = Multispectral ratio

2 = Grey scale adjustment

Option type: 1 or 2

This defines the type of enhancement used for the third display area.

If a 2 has been input for question 7, the following question is asked:

(7a) Input the type of G.S.A. to use --

1 = Standard deviation method

2 = Histogram equilization method

OPTION >> 1 or 2

This defines the type of grey scale adjustment needed.

(8) Enter element, line offset between the 2 scenes (2I4): n,n

Each of these numbers may be in the range of ±32767.

(9) Input file name for search area 2: name B2.ext

See Instruction (4).

(10) Input algorithm number(s) to use

SSDA	1
CLASSICAL	2
SSDA with threshold	3

Algorithm(s) type: n |,n...|

The user may input one, two, or all three correlation algorithms to perform the correlation.

(11) Input display area(s) correlation should take place

1 = .65 micron band

2 = .85 micron band

3 = Multispectral or G.S.A. area

Display area(s) : n |,n...|

The user may perform the correlation in any or all bands.

(12) Input the landmark sizes to be used. The largest size must appear first in the list. Maximum l.m. size = 54

Landmark sizes : n |,n...|

Up to ten landmark sizes may be input.

(13) Input l.m. sizes to be sub-pixel registered: n |,n...|

The l.m. sizes in question 12 can be used.

(14) Input l.m. sizes to have their corr. surfaces plotted: n |,n...|

The l.m. sizes in question 12 can be used.

(15) The user is then prompted to locate the landmark center using the same instructions as question 6.

From here on, all correlation is autonomous. The user can sit back. When the program finishes, printout of the run may be received by typing in > PIP CORALA.DAT/SP

2.2.2 TYPE 2 - INTERACTIVE MODE, DATA BASE STORAGE

The following is the sequence of questions asked the user when under a type 2 run:

(1) Will this run use a prestored run controller (Y or N) N

Inputting an N will cause this to be in the interactive mode.

(2) Is this run to update the corr. data base (Y or N) Y

This will put the user in the data base storage mode.

(3) Input file name for search area 1 : name B2.ext

Each image file name is designated by a name appended with a B2 or B4 signifying Band 2, .65 micron, or Band 4, .85 micron. For this question, B2 is appended. The file extension, ext, is a 3-letter file description type, usually DAT.

(4) Input starting element, line, and ratio (3I5): n,n,n

Each of these numbers may be in the range ±32767. Default display ratio is 1. The element and line inputs define the absolute location of the imagery to be displayed.

(5) The user is prompted to flip switch 1 of the joystick up. Then the user positions the cursor, by using the joystick, to the center of the desired search area. Once positioned, the user must flip switch 4 up, then down.

(6) Enter element, line offset between the 2 scenes (2I4) : n,n

Each of these numbers may be in the range of ±32767.

(7) Input file name for search area 2 : name B2.ext

See Instruction (4).

From here on, all correlation is autonomous. The use can sit back. When the program finishes, printout of the run may be received by typing in > PIP CORALA.DAT/SP

Default values used for a type 2 run:

(1) Band 3 uses multispectral ratioing.

(2) All 3 algorithms are used for the correlation.

(3) Bands 1 and 2 are used for the correlation.

(4) Landmark sizes of 48, 32, 24, 16, and 8 are used.

(5) Subpixel registration is performed on all l.m. sizes.

(6) No correlation surfaces are displayed.

2.2.3 <u>TYPE 3 - BATCH MODE, DATA BASE STORAGE</u>

The following questions are asked the user when under a type 3 run:

(1) Will this run use a prestored run controller (Y or N) Y

This will cause batch mode to become active.

(2) Is this run to update the corr. data base (Y or N) Y

This activates the data base storage mode.

From this point on, everything is autonomous. Defaults, landmark sizes, etc., are the same as those for a type 2 run.

3.0 PROGRAMMER'S SECTION

3.1 Introduction

The CORALA programs use a tree structure to define their operation. The root of the tree is the main program CORALA. This main program acts as a driver to the three major factors: CORINI, CORMID, and CORFIN. These programs in turn act as drivers to other subroutines. The tree structure may be seen in Section 3.4 (Figure 3.4b).

All programs were written in FORTRAN 4 PLUS with the exception of the I/O routine ADJUST. These programs were designed and implemented on a DEC PDP 11/70 running RSX 11-M operating system. The graphics device used was a RAMTEK 9100 320 x 256 pixel resolution graphic display system.

The CORALA subroutines are classified into three types:

(1) Program specific

(2) Library routines

(3) I/O routines

3.2 Variable Conventions

Whenever possible, the following variable naming conventions were used:

(1) Any variable beginning with an I is considered an I*2 variable (2 bytes).

(2) Any variable beginning with an R is considered an R*4 variable (4 bytes).

(3) Any variable beginning with an A or L is considered byte variable (1 byte).

(4) Any variable not following the previous conventions are declared using standard FORTRAN variable conventions.

3.3 Common Blocks

Common blocks are used extensively in the CORALA software package. These common blocks facilitate the sharing of common variables. The blocks of code are inserted in the appropriate source programs by using the INCLUDE

statement. This eliminates the possibility of a typographical error when typing in multiple common blocks.

3.3.1 DATA.COM

This common block contains all scratch arrays and various other data arrays and variables. This is sort of a catch-all common block.

```
COMMON /DATA/ IWIN(64,64),ILM(64,64),IBUF(320),ISASZE,IJOY(4),
              IESL(3),ILSL(2),IBFE,IBFL,ITBFE,ITBFL,ICLDTH,
              LBUF(512),IRATIO,IBOPT,IGSATY,RTHRES,VEGBAR,BARWAT,
              RBFE,RBFL,IRED(256),IRED1(256),ICONT
INTEGER*2     IWIN,ILM,IBUF,ISASZE,IJOY,IESL,ILSL,IBFE,IBFL,
              ITBFE,ITBFL,ICLDTH,IRATIO,IBOPT,IGSATY,IRED,
              ICONT,IRED1
REAL*4        RTHRES,RBFE,RBFL,VEGBAR,BARWAT
LOGICAL*1     LBUF,LUBF1(8192)
EQUIVALENCE   (LBUF1(1),ILM(1))


IWIN   -- SCRATCH ARRAY
ILM    -- SCRATCH ARRAY
IBUF   -- SCRATCH ARRAY
ISASZE -- SEARCH AREA SIZE IN PIXELS
IJOY   -- JOYSTICK SCRATCH BUFFER
IESL   -- ELEMENT START ARRAY
ILSL   -- LINE START ARRAY
IBFE   -- BEST FIT ELEMENT
IBFL   -- BEST FIT LINE
ITBFE  -- TEMPORARY BEST FIT ELEMENT
ITBFL  -- TEMPORARY BEST FIT LINE
ICLDTH -- VR. CLOUD THRESHOLD VALUE
```

LBUF -- SCRATCH ARRAY

IRATIO -- DISPLAY RATIO

IBOPT -- OPTION TYPE FOR THIRD DISPLAY AREA

IGSATY -- OPTION TYPE FOR GREY SCALE ADJUSTMENT

RTHRES -- CLOUD PERCENTAGE MAXIMUM VALUE

VEGBAR -- VEGETATION/BARE EARTH BOUNDARY RATIO

BARWAT -- BARE EARTH/WATER BOUNDARY RATIO

RBFE -- BEST FIT ELEMENT (REAL)

RBFL -- BEST FIT LINE (REAL)

IRED -- SCRATCH ARRAY

IRED1 -- SCRATCH ARRAY

ICONT -- AUTO CORRELATION FLAG

3.3.2 DATABS.COM

This common block contains the variables and arrays for the data base storage mode.

```
COMMON /DATABS/ IOUT(1792),IAUTO(81),IACT(81),IDBUP,IRECNO
LOGICAL*1      LOUT(3584)
REAL*4         ROUT(896)
INTEGER*2      IOUT,IAUTO,IACT,IDBUP,IRECNO
EQUIVALENCE    (IOUT(1),LOUT(1),(IOUT(1),ROUT(1)))
```

IOUT -- OUTPUT ARRAY USED TO HOLD ALL CORRELATION PERTINENT DATA

IAUTO -- ARRAY CONTAINING THE AUTO CORRELATION VALUES

IACT -- ARRAY CONTAINING THE ACTUAL CORRELATION VALUES

IDBUP -- DATA BASE UPDATE FLAG. 'Y'=UPDATE, OTHERWISE NO UPDATE

IRECNO -- CURRENT NUMBER OF TEST CASES STORED IN THE DATA BASE

3.3.3 FILES.COM

This common block contains the LANDSAT file names and the sizes of these names of the images currently displayed.

```
COMMON /FILES/ IF2SZE,FNAME1(20),FNAME2(20)

LOGICAL*1    FNAME1,FNAME2
INTEGER*2    IF1SZE,IF2SZE
```

IF1SZE -- SIZE OF THE NAME FOR THE FIRST DATA FILE

IF2SZE -- SIZE OF THE NAME FOR THE SECOND DATA FILE

FNAME1 -- ARRAY CONTAINING THE FIRST FILE NAME

FNAME2 -- ARRAY CONTAINING THE SECOND FILE NAME

3.3.4 HEADER.COM

This common block contains all the HEADER arrays used for the extraction and display of the LANDSAT image HEADER block.

```
COMMON /HEADER/ SID(11),EXPODT(7),FORCEN(14),SUNANG(21),SUNANA(3),
                NADIR(14),HEADNG(3)
LOGICAL*1       SID,EXPODT,FORCEN,SUNANG,SUNANA,NADIR,HEADNG
```

SID -- SATELLITE IDENTIFICATION

EXPODT -- EXPOSURE DATE

FORCEN -- FORMAT CENTER

SUNANG -- SUN ANGLE

SUNANA -- SUN AZIMUTH ANGLE

NADIR -- NADIR LATITUDE AND LONGITUDE

HEADNG -- SATELLITE HEADING

3.3.5 IO.COM

This common block contains all pertinent I/O variables used to talk to the RM9100.

```
COMMON /IO/ IOST(4),IPARAM(6),IRMCMD(30),IDS
```

IOST -- I/O STATUS ARRAY

IPARAM -- INSTRUCTION PARAMETER ARRAY

IRMCMD -- RAMTEK INSTRUCTION ARRAY

IDS -- DIRECTIVE STATUS

3.3.6 OPTS.COM

This common block contains the user input control options.

```
COMMON /OPTS/ ISURF(10),IBOX(10),IALG(3),IBANDS(3),ISBOX(10),
              INBOX,INBAND,INALG
INTEGER*2     ISURF,IBOX,IALG,IBANDS,ISBOX,INBOX,INBAND,INALG
```

ISURF -- CONTAINS THE LANDMARK SIZES TO HAVE THEIR CORRELATION SURFACES DISPLAYED

IBOX -- CONTAINS THE LANDMARK SIZES TO USE

IALG -- CONTAINS WHICH ALGORITHMS TO USE

IBANDS -- CONTAINS WHICH BANDS TO USE

ISBOX -- CONTAINS WHICH L.M. SIZES TO SUBPIXEL REGISTER

INBOX -- THE NUMBER OF LANDMARK SIZES INPUT

INBAND -- THE NUMBER OF BANDS INPUT

INALG -- THE NUMBER OF ALGORITHMS INPUT

3.3.7 COMMON BLOCK REFERENCE LIST

	DATA	DATABS	FILES	HEADER	IO	OPTS
ADJ	X					
AUTO	X					
AUTOCR	X	X				
CLOUDC	X					
CLOUDS	X					
CLSCOR	X					
CORALA	X	X	X	X	X	X
CORFIN	X	X				X
CORG	X					
CORINI	X	X	X	X		
CORMID	X	X				X
CORSDA	X					
CRALGS	X	X				
CRCORR		X				
DISPLE	X	X				
DIVID	X					
GETHDR				X		
GRID	X					
GSAHIS	X					
GSASTD	X					
LMDISP				X		
NXDREC	X					
POS	X					
RESET					X	
RMCHCK					X	
RMERSE	X					
RMRDIM					X	
RMTEXT					X	
RMWRIM					X	
SAREA	X	X				
SCALE	X					
SPIX	X					
WIO					X	
WRITSC	X	X	X			

3.4 CORALA Program Tree Structure

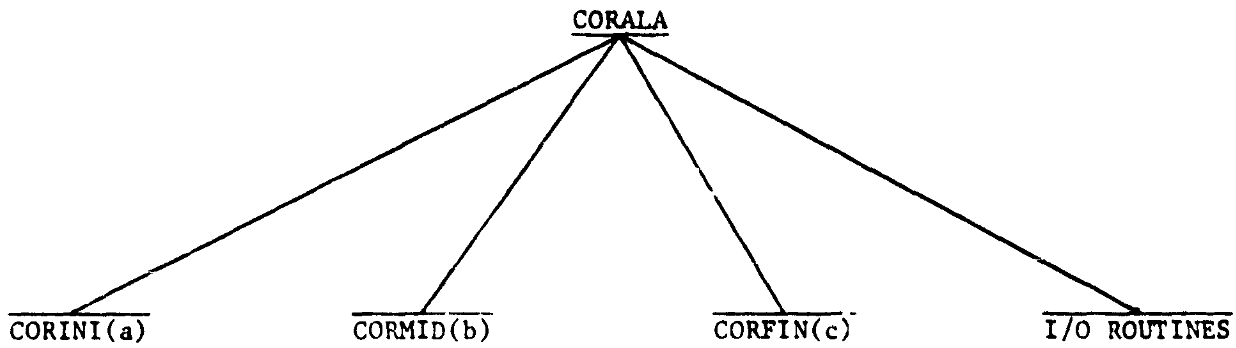


Figure 3.4a

FIGURE 3.4b

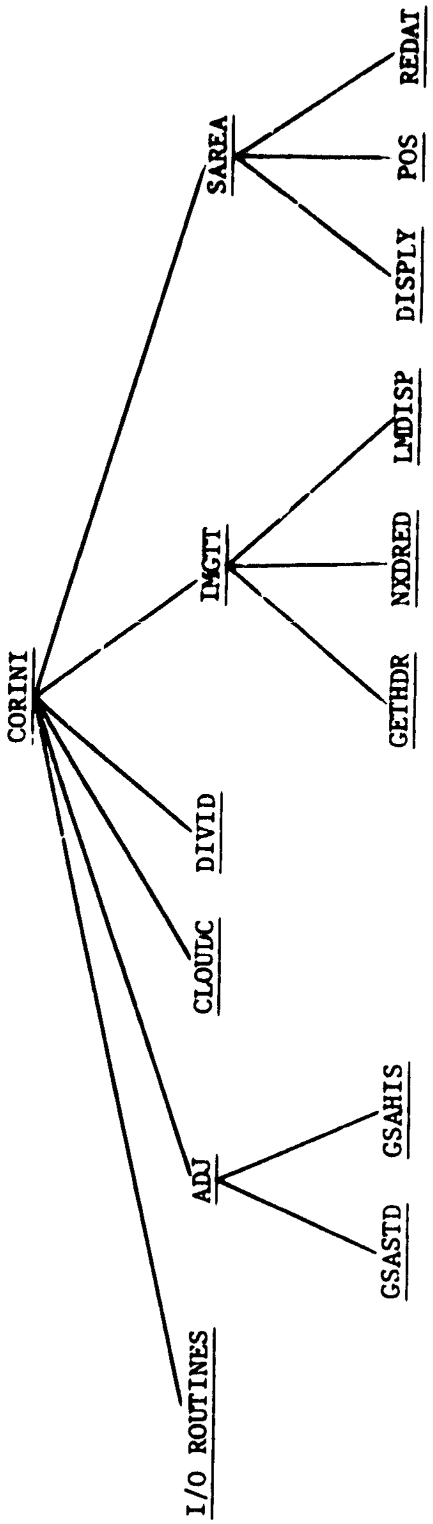
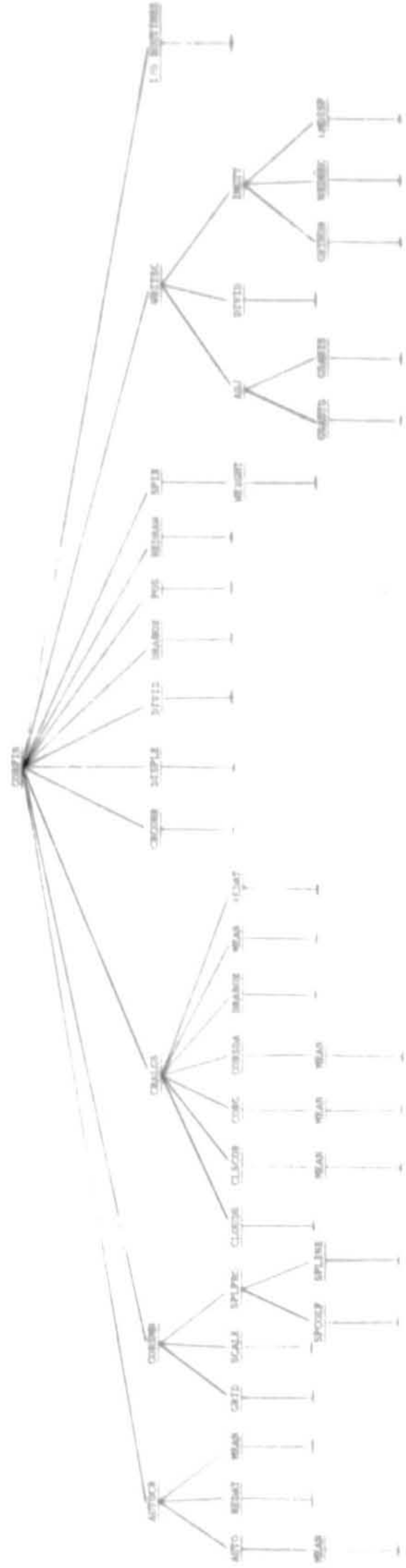


FIGURE 3.4c



3.5 Program specific routines are those routines used specifically for correlation analysis. These routines are:

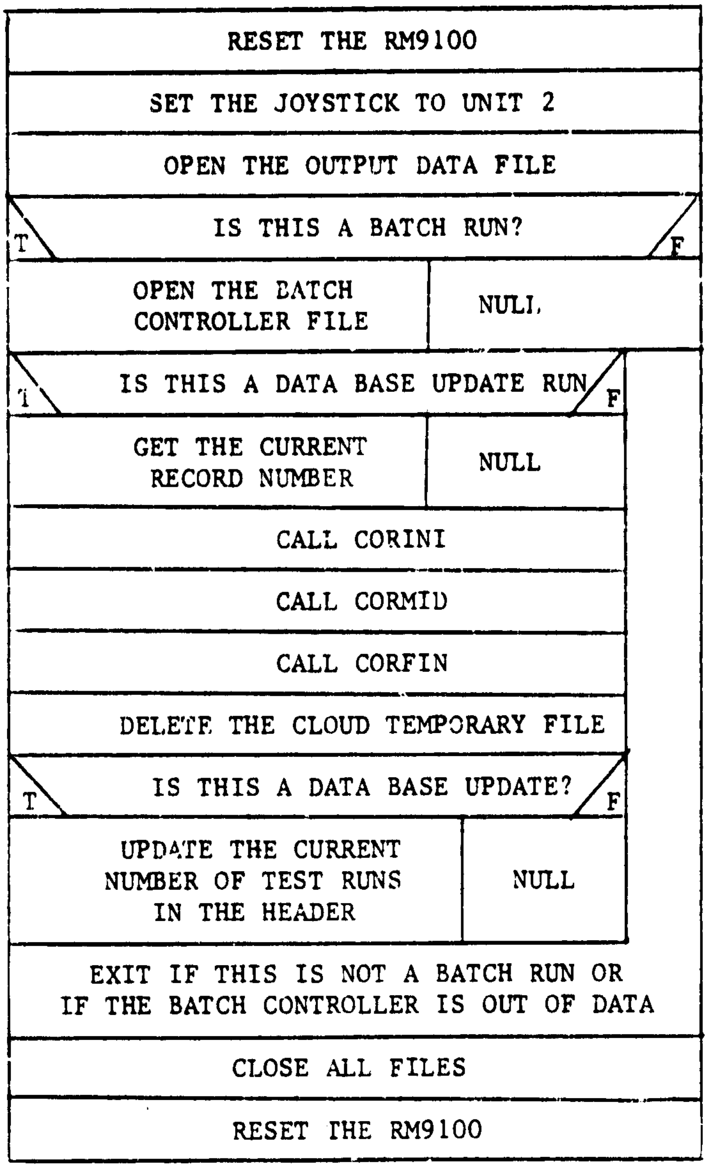
		SECTION
1.	ADJ	3.5.5
2.	AUTO	3.5.6
3.	AUTOCR	3.5.7
4.	CLOUDC	3.5.8
5.	CLOUDS	3.5.9
6.	CLSCOR	3.5.10
7.	CORALA	3.5.1
8.	CORFIN	3.5.4
9.	CORC	3.5.11
10.	CORINI	3.5.2
11.	CORMID	3.5.3
12.	CORSDA	3.5.12
13.	CORSMN	3.5.13
14.	CRALGS	3.5.14
15.	CRCORR	3.5.15
16.	DISPLE	3.5.16
17.	DISPLY	3.5.17
18.	DIVID	3.5.18
19.	GRID	3.5.19
20.	GSAHIS	3.5.20
21.	GSASTD	3.5.21
22.	SAREA	3.5.22
23.	SCALE	3.5.23
24.	SPIX	3.5.24
25.	SPLPRO	3.5.25
26.	WEIGHT	3.5.26

3.5.1 CORALA (Correlation Analysis)

CORALA is the main program for the correlation analysis programs. This program will allow the user to run the correlation programs either in a semi-batch mode or an interactive mode.

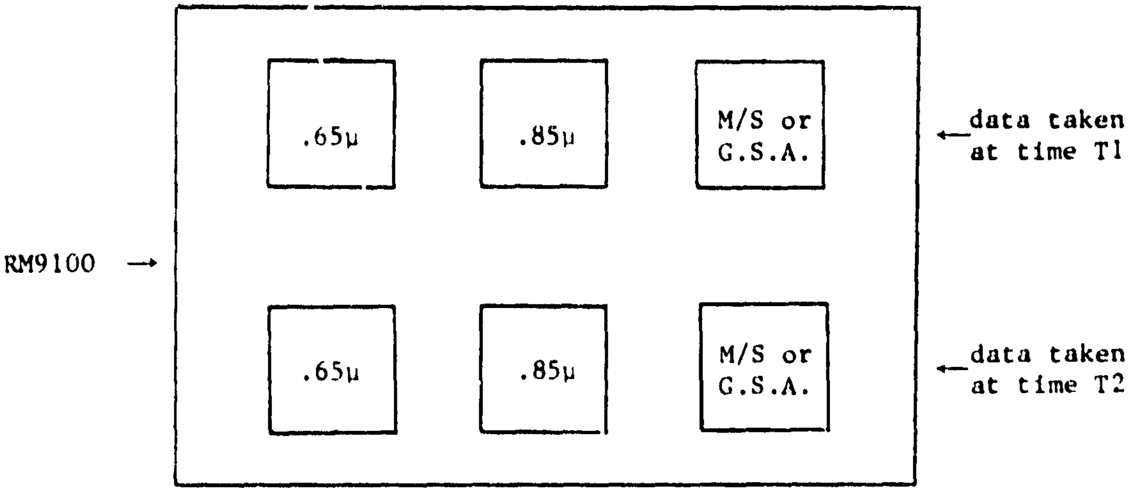
CORALA assigns the joystick to unit 2, resets the RM9100 and opens the data base and output files. This program then drives the major subroutines CORINI, CORMID, and CORFIN.

CORALA



3.5.2 CORINI (Correlation Initialization)

The CORINI routine is the initialization subroutine for the correlation analysis programs. This subroutine will display the image data in display areas 1, 2, and 3. These display areas will contain the user-defined search area taken at two different times.



This routine will call other routines that display two 320 x 120 pixel image areas of the same imagery in two different bands. The use then defines the search area size and defines the center of this search area in the imagery displayed. The user will then define which type of algorithm for display area 3 should be used. The corresponding data taken at time 2 will then be displayed. Control is then returned to CORALA.

IS THIS A DATA BASE RUN?

T: GET THE TEST CASE NUMBER FROM THE DATA BASE

F: GET THE TEST CASE NUMBER FROM THE USER

IS THIS A BATCH RUN?

T: GET THE FILE NAME FOR THE TIME 1 DATA FROM THE BATCH CONTROLLER

F: GET THE FILE NAME FOR THE TIME 1 DATA FROM THE USER

IS THIS A BATCH RUN?

T: GET THE STARTING ELEMENT, LINE, RATIO FROM THE BATCH CONTROLLER

F: GET THE STARTING ELEMENT, LINE, RATIO FROM THE USER

DISPLAY THE 2 SPECTRAL BANDS OF DATA

GET THE SEARCH AREA AND SIZE FROM THE USER

GET THE THIRD AREA DISPLAY TYPE AND DISPLAY

IS THIS A BATCH RUN?

T: GET THE SECOND FILE NAME FROM THE BATCH CONTROLLER

F: GET THE SECOND FILE NAME FROM THE USER

DISPLAY THE LOWER HALF OF THE SCREEN WITH TIME 2 IMAGE DATA

CREATE THE CLOUD DATA FILE SPECIFYING WHICH PIXELS IN THE SECOND SET OF IMAGERY ARE CLOUDS

3.5.3 CORMID (Correlation Middle Routine)

The CORMID subroutine queries the user to input certain control parameters. In a data base storage mode, these parameters are pre-defined. The questions asked are:

(1) Algorithm(s) to be used for correlation

(2) Band(s) for the correlation to be tested

(3) Landmark size(s) to be tested

(4) Landmark size(s) to be subpixel registered

(5) Landmark size(s) to have correlation surface plotted

The default values used for a data base storage mode are:

(1) S.S.D.A. and classical correlation

(2) Bands 1 and 2

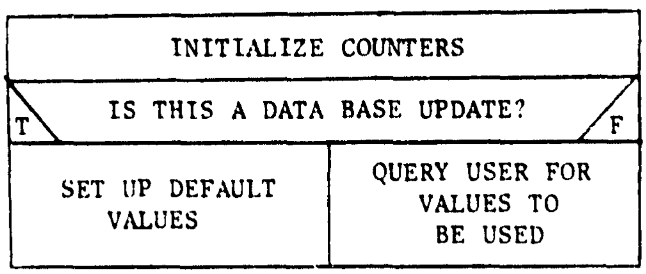
(3) Landmark sizes of 48, 32, 24, 16, and 8

(4) Subpixel landmark sizes of 48, 32, 24, 16, and 8

(5) No correlation surfaces plotted

FORMAT: CALL CORMID

CORMID



3.5.4 CORFIN (Correlation Finish)

Once all the input values and image data are received from the user, CORFIN proceeds to perform the actual correlation analysis. This subroutine drives the analysis programs.

Hard-copy correlation results are printed from this routine. This routine also stores the correlation results into the data base if this is a data base mode run.

FORMAT: CALL CORFIN

CORFIN

IS THIS A BATCH MODE RUN?

T: GET LANDMARK LOCATION FROM THE BATCH FILE

F: GET LANDMARK LOCATION BY ALLOWING THE USER TO USE THE JOYSTICK

- DO FOR THE NUMBER OF BANDS SPECIFIED
  - DO FOR THE NUMBER OF ALGORITHMS SPECIFIED
    - DO FOR THE LANDMARK SIZES SPECIFIED
      - PERFORM AUTO CORRELATION
      - DRAW A BOX AROUND THE REFERENCE LANDMARK
      - SET UP TIMER TO ZERO AND START TIMING
      - CALL THE ROUTINE TO PERFORM CORRELATION
      - STOP TIMER AND STORE THE VALUE
      - DETERMINE THE LANDMARK CLOUD PERCENTAGES
      - DETERMINE THE CORRELATION COEFFICIENT
      - DISPLAY THE CORRELATION SURFACE IF DESIRED
      - PERFORM SUBPIXEL REGISTRATION
      - RESTORE SCREEN
      - OUTPUT DATA

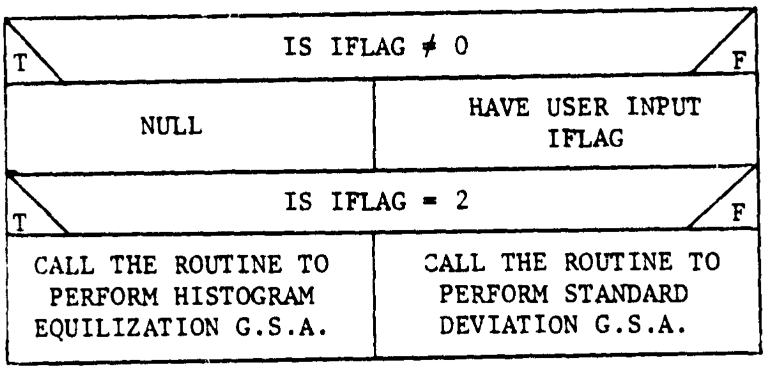
3.5.5 <u>ADJ</u> (Adjust data)

ADJ is the driver routine used for grey scale adjustment of image data. Two methods of grey scale adjustment are currently implemented. One method is the standard deviation method. The other is the histogram equilization method. ADJ allows the user to specify the appropriate method and then calls the subroutine that implements this method.

Two flags are passed to this subroutine. IFLAG is a decision flag declaring which method of grey scale adjustment is to be used. If IFLAG equals zero, the user is prompted for the G.S.A. type. IFLAG2 is another decision flag used for hard-copy output. This variable will be used by the adjustment routines. ILS is the line start of the imagery.

FORMAT: CALL ADJ (ILS,IFLAG,IFLAG2)

<u>ADJ</u>



3.5.6 AUTO (Auto Correlation)

AUTO is used in conjunction with the subroutine AUTOCR to perform auto correlation. This subroutine will be called 81 times by AUTOCR to create a 9 x 9 element auto correlation surface.

This routine performs the correlation algorithm on one placement. The algorithm type is defined by ITALG. ITAUTO is the correlation value determined by this placement. IX and IY are offsets in the array IWIN which contains the landmark image data. RLMEAN is the landmark mean and ISIZE is the landmark size. ICORR is used by the S.S.D.A. with thresholding algorithm and contains the current minimum correlation value.

FORMAT: CALL AUTO(ITAUTO,IX,IY,RLMEAN,ISIZE,ITALG,ICORR)

AUTO

INITIALIZE CORRELATION VALUE TO 0		
ALGORITHM TYPE		
SSDA	CLASSICAL	SSDA to Threshold
Correlation value = $\sum\sum abs(win - \overline{win}) - (lm - \overline{lm})$	Correlation value = $\dfrac{\sum\sum(win - \overline{win})*(lm - \overline{lm})}{(\sum\sum(win-\overline{win})^2 * \sum\sum(lm-\overline{lm})^2)^{1/2}}$ SCALE CORRELATION VALUE	Same as SSDA except with thresholding checking
STORE THE CORRELATION VALUE INTO AN INTEGER VARIABLE		

3.5.7 <u>AUTOCR</u> (Auto Correlation Driver)

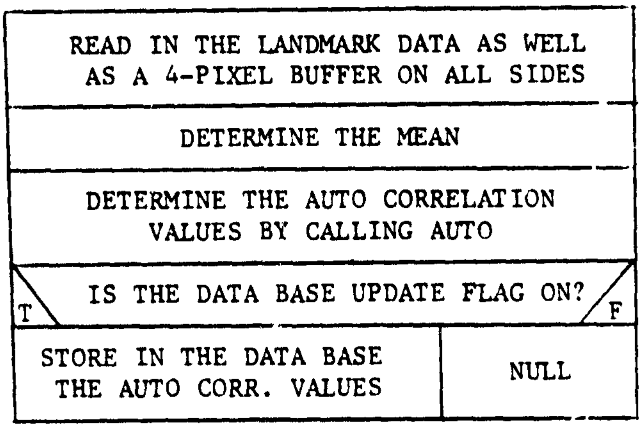
AUTOCR is the auto correlation driver routine. Auto correlation is the process of correlating a defined landmark upon itself using a 4-element offset in all directions. What this will do is describe the best possible correlation surface for this landmark. The placement values are stored in a 9 x 9 element array called IAUTO.

ISTE and ISTL are the starting screen coordinates of the landmark. ILMSZE is the landmark size and ITALG is the classification algorithm being used.

This subroutine will store the auto correlation values into the data base if the data base option is defined.

FORMAT: CALL AUTOCR(ISTE,ISTL,ILMSZE,ITALG)

<u>AUTOCR</u>

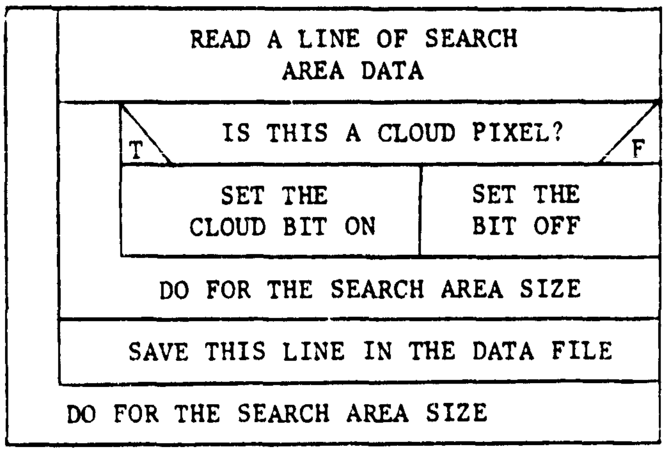


3.5.8 CLOUDC (Create Cloud File)

This subroutine takes a pixel-by-pixel ratio of two bands of one scene and creates a cloud classification array and stores this array in the data file CLOUDS.TMP. If a pixel is classified as a cloud, the sign bit is set in the cloud array. If this isn't a cloud pixel, the array element is set to zero.

FORMAT: CALL CLOUDC

CLOUDC

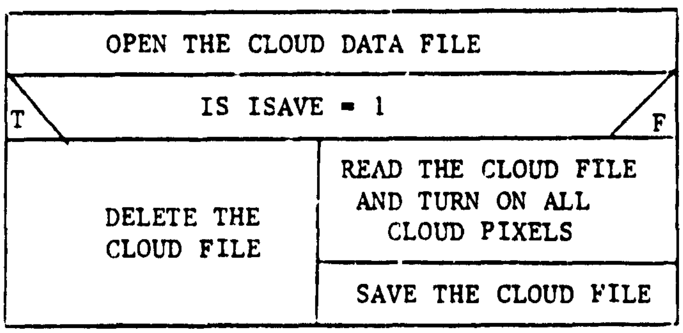


3.5.9 <u>CLOUDS</u> (Set Cloud Bit On)

This subroutine will read the cloud classification data created by CLOUDC. The displayed search area data is contained in IWIN. This subroutine will turn on any cloud pixels in this search area array. By turning the sign bit on, none of the image data is lost. This subroutine is used in conjunction with the correlation algorithms. This allows for an easy means to distinguish clouds and to not use them in the correlation.

FORMAT: CALL CLOUDS(ISAVE)

<u>CLOUDS</u>



3.5.10 CLSCOR (Classical Correlator)

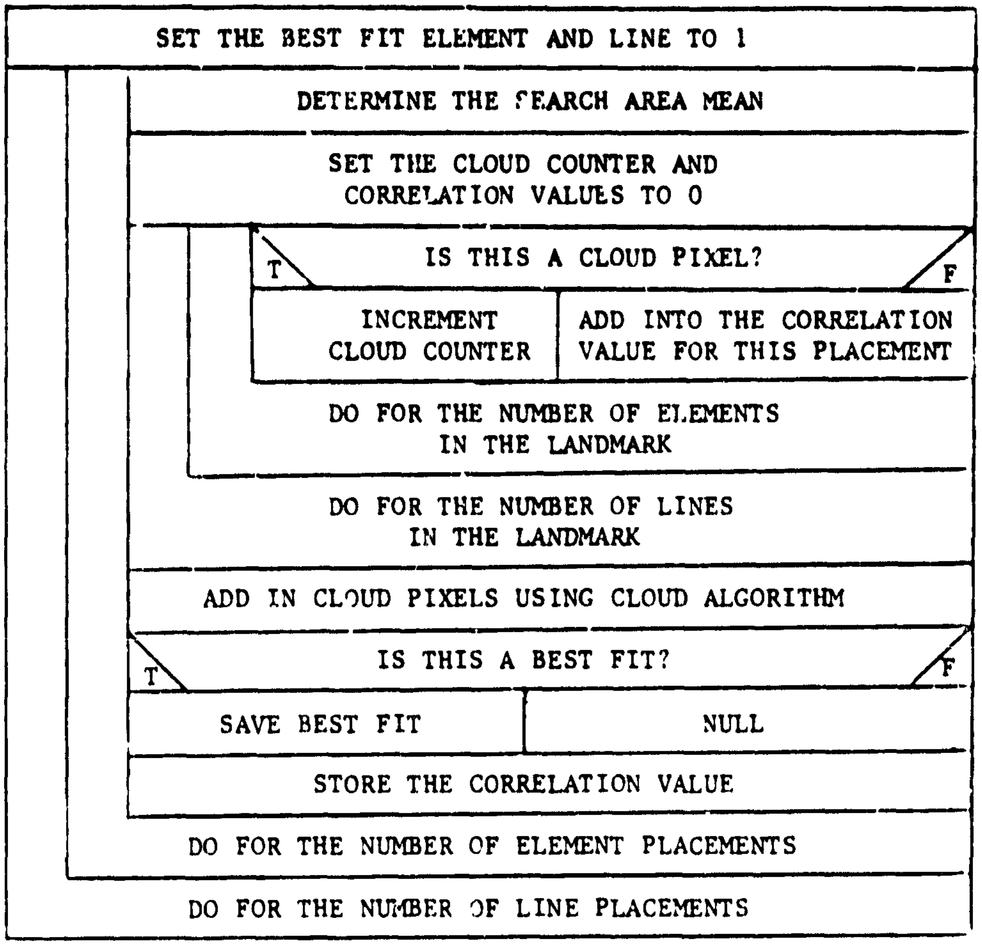
This subroutine calculates the classical correlator values for all landmark placements. The array IWIN contains the search area data. The array ILM contains the landmark data. RLMEAN is the landmark mean and ILMSZE is the landmark size. The classical correlator algorithm is:

$$\frac{\displaystyle\sum_{i=1}^{\#lines}\ \sum_{j=1}^{\#elements} \left(IWIN(J,I) - IWMEAN\right) * \left(ILM(J,I) - ILMEAN\right)}{\left(\displaystyle\sum_{i=1}^{\#lines}\ \sum_{j=1}^{\#elements} \left(IWIN(J,I) - IWMEAN\right)^2 * \sum_{i=1}^{\#lines}\ \sum_{j=1}^{\#elements} \left(ILM(J,I) - ILMEAN\right)^2\right)^{1/2}}$$

This subroutine performs an exhaustive search of the landmark data versus the search area data. The best fit is where the correlation value is at a maximum. The generated correlation values have their range between +1 and -1. These values are multiplied by 32767 and stored back into IWIN.

FORMAT: CALL CLSCOR (RLMEAN,ILMSZE)

CLSCOR


SET THE BEST FIT ELEMENT AND LINE TO 1
DETERMINE THE SEARCH AREA MEAN
SET THE CLOUD COUNTER AND CORRELATION VALUES TO 0
IS THIS A CLOUD PIXEL?
T
F
INCREMENT CLOUD COUNTER
ADD INTO THE CORRELATION VALUE FOR THIS PLACEMENT
DO FOR THE NUMBER OF ELEMENTS IN THE LANDMARK
DO FOR THE NUMBER OF LINES IN THE LANDMARK
ADD IN CLOUD PIXELS USING CLOUD ALGORITHM
IS THIS A BEST FIT?
T
F
SAVE BEST FIT
NULL
STORE THE CORRELATION VALUE
DO FOR THE NUMBER OF ELEMENT PLACEMENTS
DO FOR THE NUMBER OF LINE PLACEMENTS

3.5.11 CORG (Correlation Using SSDA)

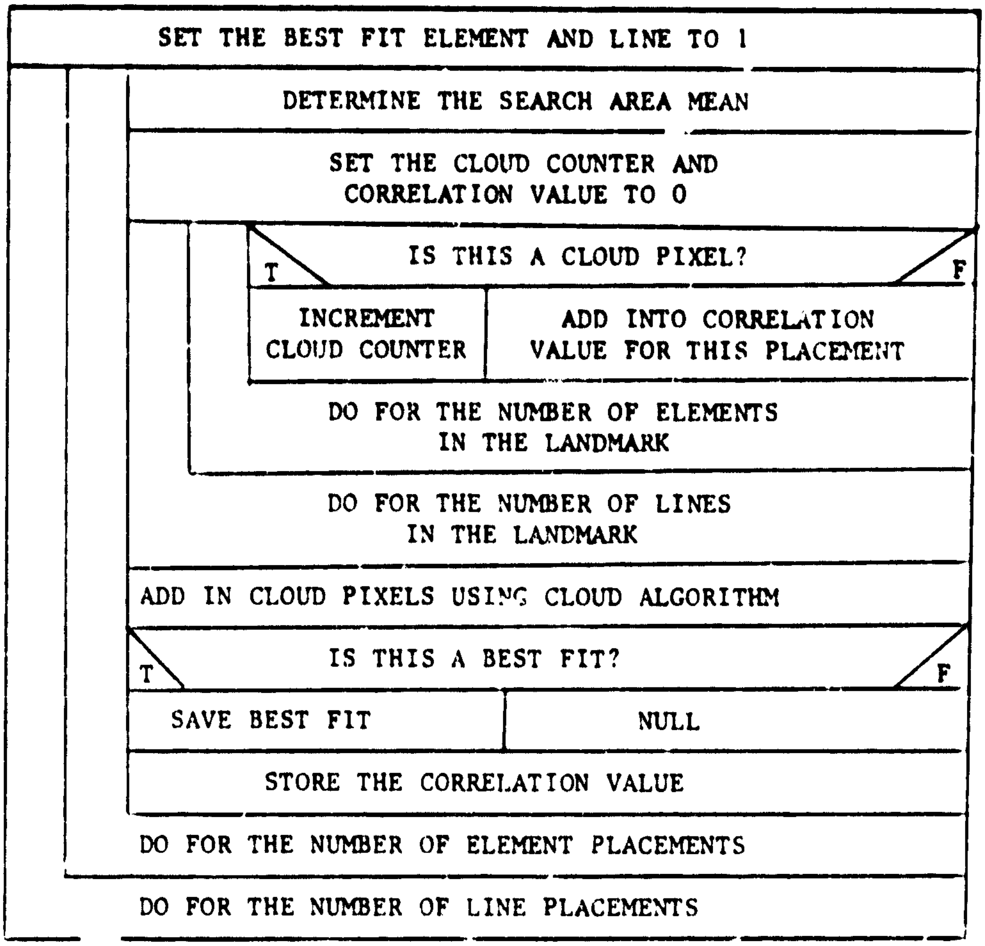
This subroutine calculates the S.S.D.A. correlation values for all landmark placements. The array IWIN contains the search area data. The array ILM contains the landmark data. RLMEAN is the landmark mean and ILMSZE is the landmark size. The S.S.D.A. algorithm is defined:

$$\sum_{i=1}^{\#\text{lines}} \sum_{j=1}^{\#\text{elements}} \text{abs}\Big(\big(\text{IWIN}(J,I) - \text{IWMEAN}\big) - \big(\text{ILM}(J,I) - \text{ILMEAN}\big)\Big)$$

This subroutine performs an exhaustive search of the landmark data versus the search area data. The best fit is where the correlation value determined by the S.S.D.A. algorithm is at a minimum. The generated correlation values are restored into IWIN.

FORMAT: CALL CORG (RLMEAN,ILMSZE)

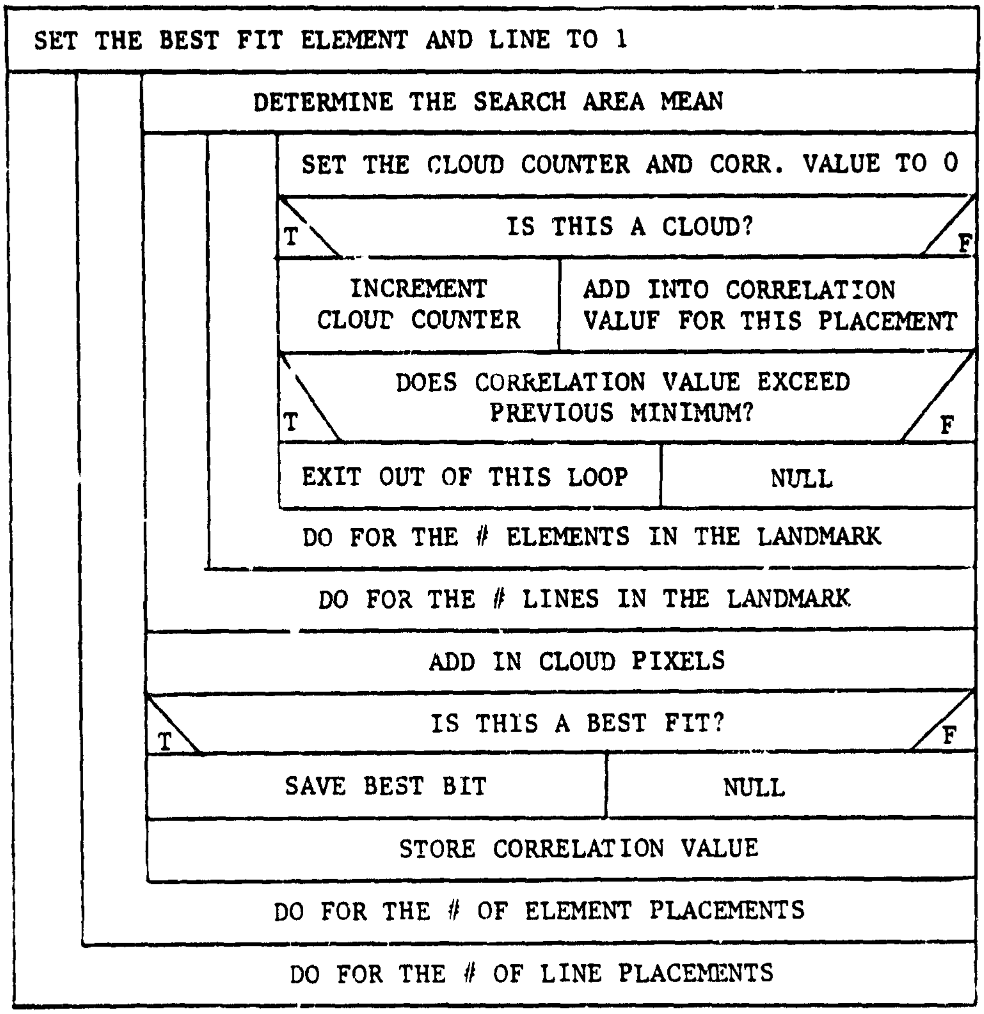
CORG


SET THE BEST FIT ELEMENT AND LINE TO 1
DETERMINE THE SEARCH AREA MEAN
SET THE CLOUD COUNTER AND CORRELATION VALUE TO 0
IS THIS A CLOUD PIXEL?
T
F
INCREMENT CLOUD COUNTER
ADD INTO CORRELATION VALUE FOR THIS PLACEMENT
DO FOR THE NUMBER OF ELEMENTS IN THE LANDMARK
DO FOR THE NUMBER OF LINES IN THE LANDMARK
ADD IN CLOUD PIXELS USING CLOUD ALGORITHM
IS THIS A BEST FIT?
T
F
SAVE BEST FIT
NULL
STORE THE CORRELATION VALUE
DO FOR THE NUMBER OF ELEMENT PLACEMENTS
DO FOR THE NUMBER OF LINE PLACEMENTS

3.5.12 CORSDA (Correlation Using S.S.D.A. $\overline{W}$ Thresholding)

This subroutine is exactly the same as CORG with the exception that threshold checking is performed. When a landmark placement is being computed, if the partial correlation value exceeds the previous minimum value, the correlation for this placement is terminated and the next placement is tested.

CORSDA



3.5.13 <u>CORSMN</u> (Correlation Surface Main Driver)

CORSMN is the driver program to display a correlation surface. This routine calls three routines: GRID, SCALE, and SPLPRO.

FORMAT: CALL CORSMN

<u>CORSMN</u>

CALL THE GRID ROUTINE
CALL THE SCALING ROUTINE
CALL THE DISPLAY ROUTINE

3.5.14 CRALGS (Correlation Algorithm Controller)

CRALGS is the driver to the various correlation algorithms. This subroutine is called by CORALA. This subroutine will call the appropriate algorithm based upon the variable ITALG. ILMSZE is the current landmark size and IES,ILS are the screen coordinates of the upper left hand corner of the current landmark. IES3 is the search area's starting screen coordinate.

Upon return from the various algorithm routines, this subroutine will draw a box around the best fit placement and will store the actual correlation values into the array IACT. IACT will be stored into the data base if the data base mode is on.

FORMAT: CALL CRALGS(ILMSZE,IES,ILS,ITALG,IES3)

CRALGS

READ IN THE SEARCH AREA DATA IN AN ARRAY
READ IN THE LANDMARK DATA
CALL THE ROUTINE TO TURN ON CLOUD BITS
GET THE LANDMARK MEAN
CALL THE CORRECT CORRELATION ALGORITHM
DRAW A BOX AROUND THE BEST FIT
STORE THE CORRELATION VALUES

3.5.15 CRCORR (Create Correlation Coefficient)

The CRCORR subroutine creates the correlation coefficient of the auto correlation surface versus the actual computed correlation surface. This value is between 1. A good correlation is one with a high positive correlation coefficient. RCORR is the value of the correlation coefficient. Algorithm used:

$$\frac{\sum_{I=1}^{81} \left(IAUTO(I) - \overline{IAUTO(I)}\right)*\left(IACT(I) - \overline{IACT(I)}\right)}{\left(\sum_{I=1}^{81} \left(IAUTO(I) - \overline{IAUTO(I)}\right)^2*\sum_{I=1}^{81} \left(IACT(I) - \overline{IACT(I)}\right)^2\right)^{1/2}}$$

FORMAT: CALL CRCORR (RCORR)

3.5.16 DISPLE (Display the Slope)

When the user specifies that a correlation surface should be displayed, DISPLE will display the best fit's correlation value along with a one placement buffer on all sides. This displays a 3 x 3 matrix of correlation values. Both the actual and the auto correlation values are displayed in the RM9100.

FORMAT: CALL DISPLE

3.5.17 DISPLY (Display Imagery)

The DISPLY subroutine displays an M x N matrix of picture elements on the RM9100 beginning at line ILS and element IES. The array IARRAY contains the data to be displayed. This array contains data in an IARRAY (element #, line #) format. IELEMS specifies the number of elements and ILINES the number of lines to be displayed.

FORMAT: CALL DISPLY(IARRAY,IBUF,IELEMS,ILINES,IES,ILS)

DISPLY

GET A LINE OF DATA IN
IARRAY STORING INTO IBUF

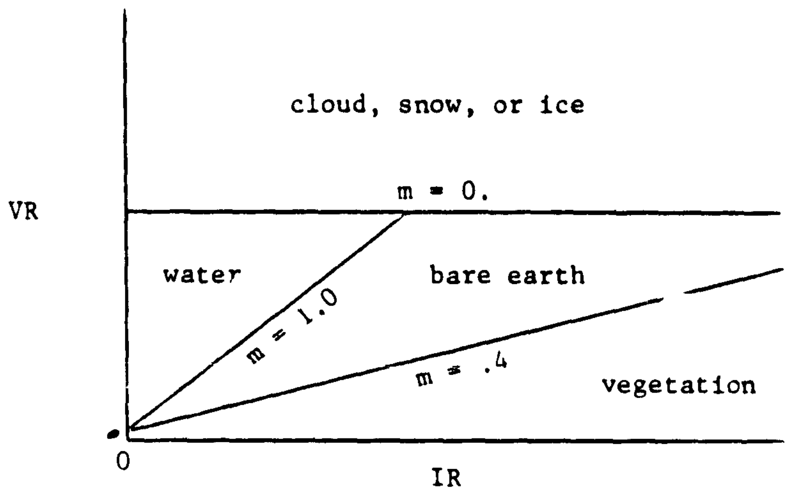
WRITE IBUF TO THE RM9100

DO FOR THE NUMBER OF LINES

3.5.18 DIVID (Classify the Image)

DIVID is used to create a classified image using an image's .65 micron and .85 micron data. This routine will classify data to four categories: 1) water; 2) vegetation; 3) bare earth; and 4) clouds, snow, or ice.

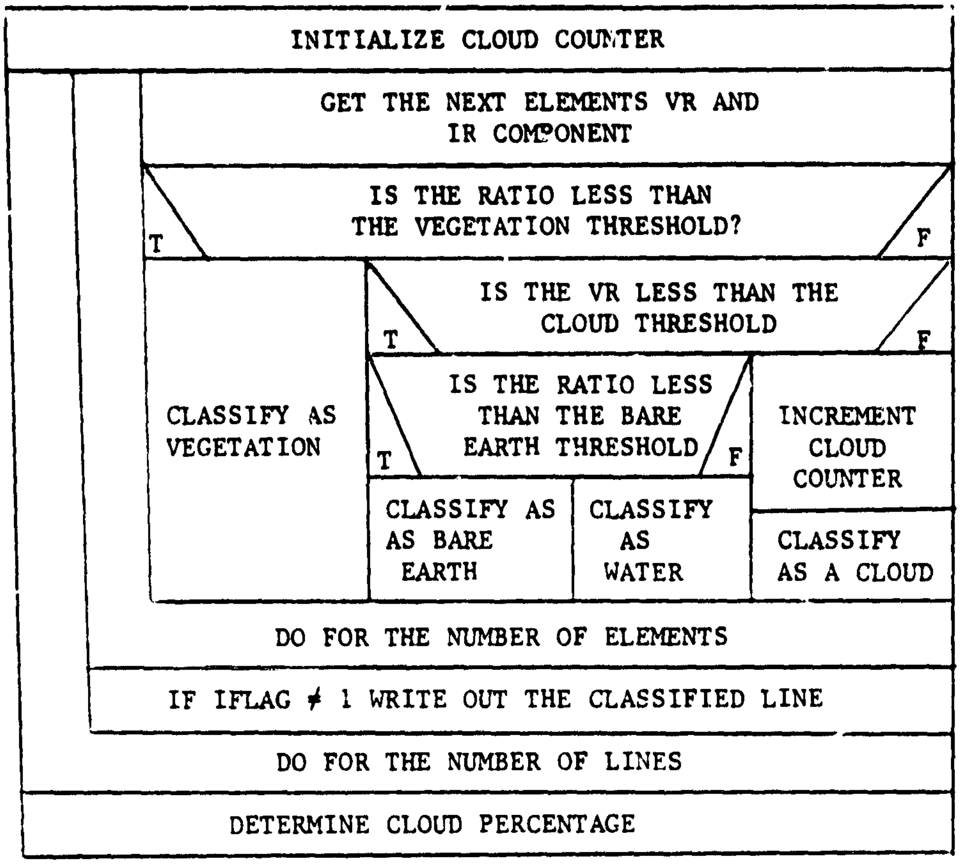
A ratioing algorithm is used for this classification. The VR/IR ratio defines the classification.



This routine will classify a square ISIZE x ISIZE area. The .65 and .85 micron data must currently be displayed on the RM9100. The point (IES1,ILS) defines the upper left corner of the .65 micron data and the point (IES2,ILS) the corresponding .85 micron data location. The point (IES3, ILS) is the upper left hand corner of the display area the classified image will be written to. The classified image is written out in 4 grey levels--25, 50, 75, and 100 corresponding to water; vegetation; bare earth; and clouds, snow, or ice, respectively. IFLAG flags whether the classified image is to be written to the RM9100. If IFLAG equals one, the data is not written; otherwise it is. ICLOUD is the percentage of cloud pixels in this scene.

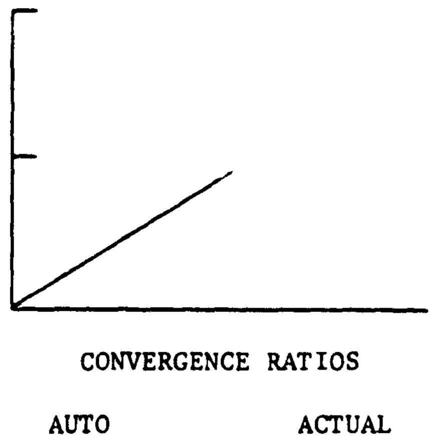
FORMAT: CALL DIVID(ISIZE,ILS,IES1,IES2,IES3,IFLAG,ICLOUD)

DIVID


INITIALIZE CLOUD COUNTER
GET THE NEXT ELEMENTS VR AND IR COMPONENT
IS THE RATIO LESS THAN THE VEGETATION THRESHOLD?
T
F
CLASSIFY AS VEGETATION
IS THE VR LESS THAN THE CLOUD THRESHOLD
T
F
IS THE RATIO LESS THAN THE BARE EARTH THRESHOLD
T
F
INCREMENT CLOUD COUNTER
CLASSIFY AS AS BARE EARTH
CLASSIFY AS WATER
CLASSIFY AS A CLOUD
DO FOR THE NUMBER OF ELEMENTS
IF IFLAG ≠ 1 WRITE OUT THE CLASSIFIED LINE
DO FOR THE NUMBER OF LINES
DETERMINE CLOUD PERCENTAGE

3.5.19 GRID (Correlation Surface Grid)

GRID is used with the correlation surface routines. This subroutine draws on the RM9100 a 3-D type grid on which the correlation surface will be plotted. The final output of GRID is the following:



FORMAT: CALL GRID

GRID

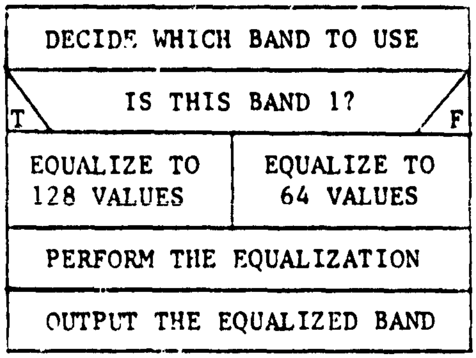
RESET THE RM9100
DRAW THE HASH MARKS
DRAW THE X AXIS
DRAW THE Y AXIS
DRAW THE Z AXIS
LABEL THE X AXIS

3.5.20 GSAHIS

This algorithm attempts to equalize the output data intensities so that the entire range of grey scales are used. The range 0 - 127 or 0 - 63 is used by the output data. This program creates the cumulative probability function and uses this for display purposes.

FORMAT: CALL GSAHIS (ILS,IFLAG2)

GSAHIS

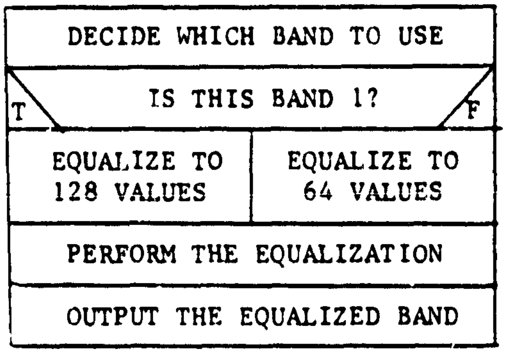


3.5.21 GSASTD

This algorithm uses a standard deviation method for grey scale adjustment of imagery. The data is adjusted to 128 or 64 grey scale values. The data is scaled using the standard deviation of the data and the number of output intensities the user desires.

FORMAT: CALL GSASTD (ILS,IFLAG2)

GSASTD



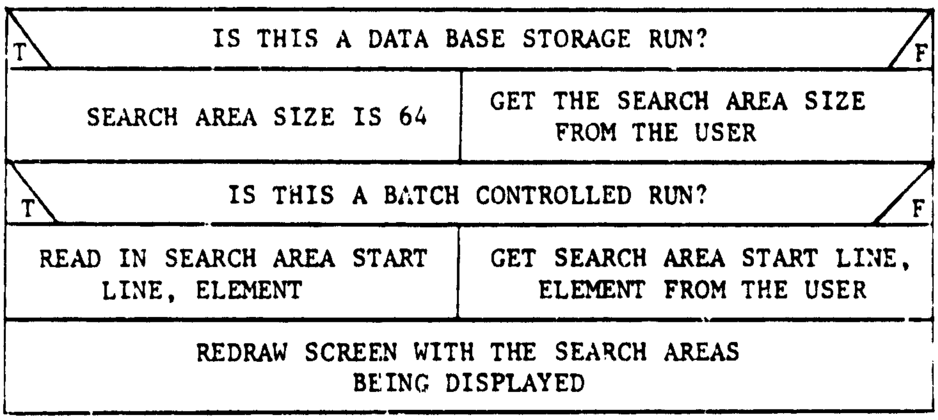
3.5.22 <u>SAREA</u> (Search Area Extraction)

This routine allows the user to move a cursor over the scene displayed on the monitor and select a search area to be used for correlation purposes. When this is done, the original scene is erased and two representations of the area are displayed on the upper portion of the screen. These representations are the two spectral scenes.

ILS is the starting line that the search area should be displayed at once they are defined by the user. IX and IY upon input are the starting coordinates for the cursor. On output, they are the upper left hand corner coordinates of the defined search area.

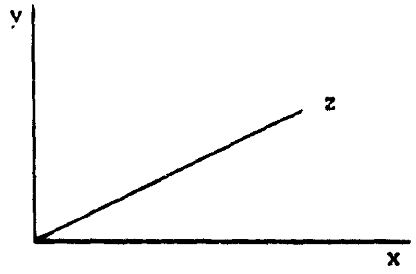
FORMAT: CALL SAREA (ILS,IX,IY)

<u>SAREA</u>



3.5 23 SCALE (Correlation Surface Scaling)

The SCALE subroutine scales the correlation surface values to fit to a grid system defined by the GRID subroutine. The GRID routine defines a 180 x 180 pixel x,y grid. A perspective view is given by defining a z axis.



IXREF is an array that will contain the scaled x points, and IYREF will contain the scaled y points of a 9 x 9 correlation surface. The best fit's correlation value will always be at

{IXREF(5,5)IYREF(5,5)}.

The correlation surface will contain the best fit and a 4-placement buffer on all sides.

SCALE

FIND THE MAXIMUM VALUE TO SCALE TO

CREATE THE X SCALE ARRAY

CREATE THE Y SCALE ARRAY

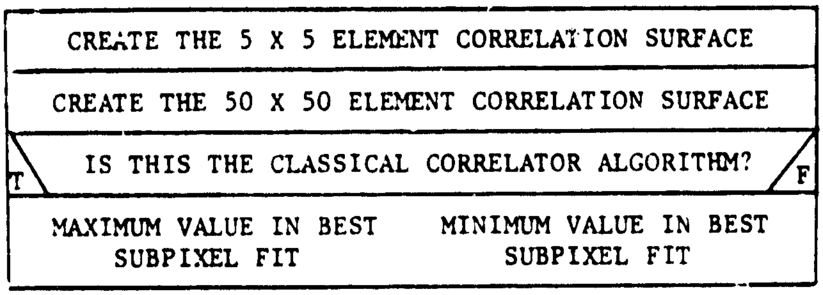
3.5.24 SPIX

This subroutine performs tenth-of-a-pixel registration of imagery. This routine used a cubic convolution process to perform the interpolation.

The array ITWIN contains the best fit correlation value along with a 2-placement buffer on all sides. This array is expanded to a 50 x 50 element array using the convolution algorithm. The best tenth-of-a-pixel fit can then be determined.

FORMAT: CALL SPIX (ITALG,ILMSZE)

SPIX



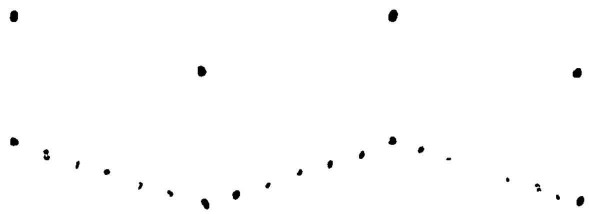
3.5.25 SPLPRO (Spline Processing)

SPLPRO is the main processing routine for the correlation surface display. This routine basically takes a 9 x 9 pixel correlation surface and interpolates this to a 180 x 180 pixel surface. The interpolation uses a cubic polynomial spline fit. This routine basically fills in the holes in the correlation surface.

EXAMPLE:

4 pt. line

20 pt. interpolated line



This routine first processes the horizontal splines and the vertical splines.

FORMAT: CALL SPLPRO (IXREF,IYREF)

SPLPRO

PROCESS HORIZONTAL SPLINES
PROCESS VERTICAL SPLINES

3.5.26 <u>WEIGHT</u> (Determine Weighting Factor)

The WEIGHT function determines the weighting factor to be used in a cubic convolution subpixel registration. The algorithm used is as follows:

$$\text{weight} = 1 - 2|x|^2 + |x|^3 \text{ if } 0 \leq |x| \leq 1$$

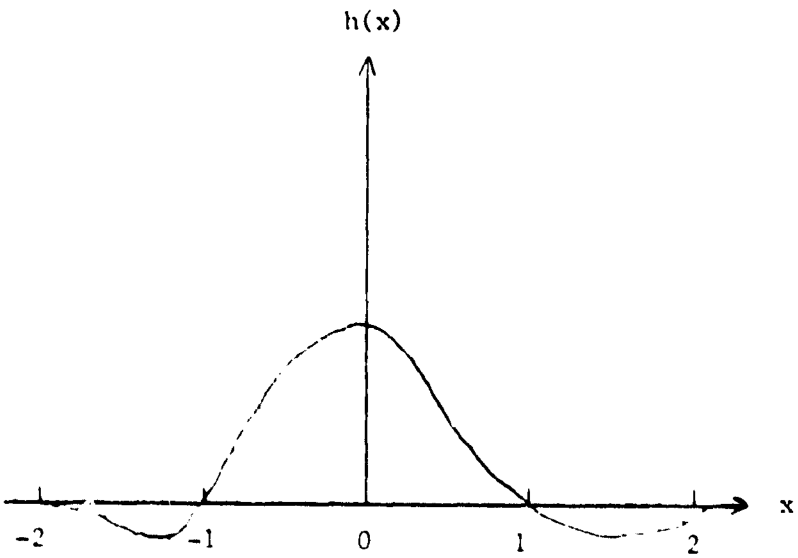
$$\text{weight} = 4 - 8|x| + 5|x|^2 - |x|^3 \text{ if } 1 \leq |x| \leq 2$$

RDEL is the x value in the equations.

FORMAT: y = WEIGHT(RDEL)

<u>WEIGHT</u>

GET ABSOLUTE VALUE OF RDEL	
IS RDEL > 1?	
WEIGHT = 4-8*RDEL+ 5**RDEL**2-RDEL**3	WEIGHT = 1-2*RDEL+ RDEL**3



3.6 Library Routines

The library routines are those subroutines which perform a general function without being strictly I/O routines. These routines are:

		SECTION
1.	DECDE	3.6.1
2.	DRABOX	3.6.2
3.	GETHDR	3.6.3
4.	IMGTT	3.6.4
5.	LMDISP	3.6.5
6.	MEAN	3.6.6
7.	NXDREC	3.6.7
8.	POS	3.6.8
9.	REDAT	3.6.9
10.	REDRAW	3.6.10
11.	SPCOEF/SPLINE	3.6.11

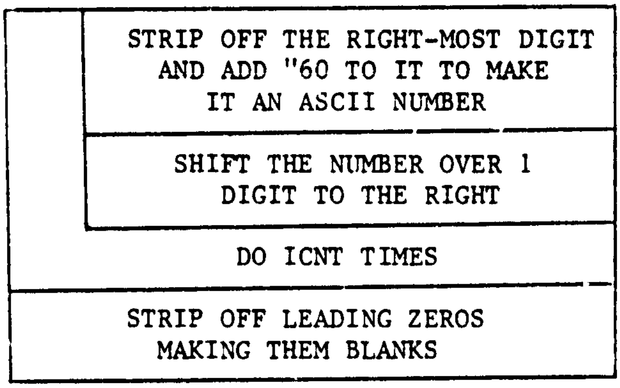
3.6.1 DECDE (Decode)

The DECDE subroutine works much like a FORTRAN DECDE statement. This routine allows the user to decode an integer into an ASCII array.

ABUF is the byte array to contain the decoded ASCII string. INUM is the integer to be decoded. ICNT is the maximum number of characters in INUM.

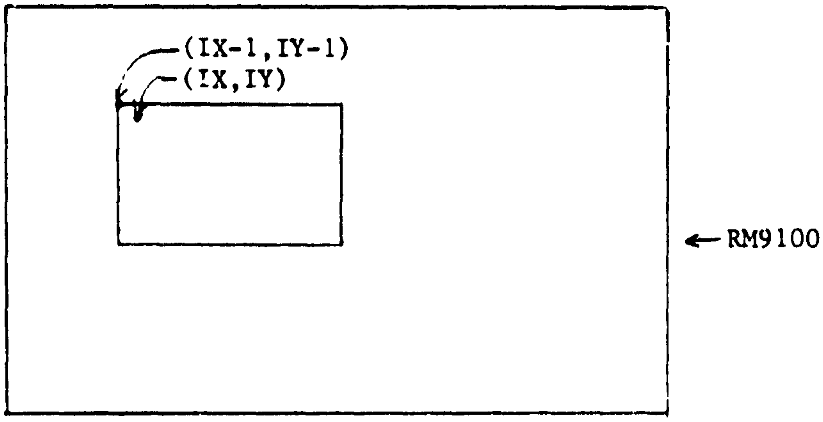
FORMAT: CALL DECDE (ABUF,INUM,ICNT)

DECDE



3.6.2 <u>DRABOX</u> (Draw a Box)

The DRABOX routine draws a square box sized ISIZE x ISIZE. INTENSE defines the intensity that this box is to be written at on the RM9100. IX and IY define the upper left hand corner of the data to be enclosed by the box. IBUF is a scratch array and IBUF1 is the array of data that the box overwrites.



FORMAT: CALL DRABOX(INTENS,ISIZE,IX,IY,IBUF,IBUF1)

Restrictions: $0 \leq$ ISIZE ≤ 62

<u>DRABOX</u>

INITIALIZE THE SCRATCH ARRAY TO THE OUTPUT INTENSITY
READ THE IMAGERY TO BE OVERWRITTEN AND SAVE IN IBUF1
WRITE THE BOX ON THE RM9100

3.6.3 GETHDR (Get Header Data)

The GETHDR subroutine gets the header data from a LANDSAT image disk file. This routine requires that unit 3 has been opened as the unit specifying LANDSAT imagery. This header data is found at record 1. The common block HEADER contains the arrays to receive this header data. The following items are retrieved:

SID(11) - 11 character satellite I.D.

EXPODT(7) - 7 character exposure date

FORCEN(14) - 14 character format center

SUNANG(2) - 2 character sun angle

SUNANA(3) - 3 character sun azimuth

NADIR(14) - 14 character nadir

LBUF is a scratch array used to read the header record.

FORMAT: CALL GETHDR (LBUF)

GETHDR

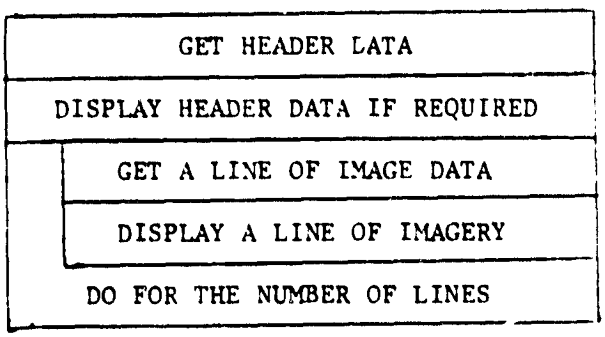
READ THE HEADER RECORD
EXTRACT AND STORE THE SID
EXTRACT AND STORE THE EXPODT
EXTRACT AND STORE THE FORCEN
EXTRACT AND STORE THE SUNANG
EXTRACT AND STORE THE SUNANA
EXTRACT AND STORE THE NADIR

3.6.4 IMGTT (Get the Image and Display)

IMGTT is the driver routine to display LANDSAT imagery onto the RM9100. This routine allows the programmer to display an M x N pixel area on the RAMTEK. IES and ILS are the starting element, line of the LANDSAT imagery to be displayed. This data is stored in a 3584 element x 2340 line data file. IESCRN and ILSCRN are the starting element, line screen coordinates at where this data will be written. NELEMS is the number of elements/lines to be written and NLINES the number of lines. IRATIO defines the output ratio of the displayed data. If IRATIO is 1, then every pixel is displayed. If IRATIO is 2, then every other pixel on every other line is displayed and so on. This gives the effect of shrinking down the imagery. IBUF and LBUF are scratch arrays. IFLAG is a flag specifying whether the header data is to be displayed on the screen. If IFLAG is greater then or equal to 1, this header information is not displayed.

```
FORMAT:  CALL IMGTT(IES,ILS,IESCREN,ILSCRN,NELEMS,NLINES,IRATIO,IFLAG,
                    IBUF,LBUF)
```

IMGTT



3.6.5 LMDISP (Display Header)

LMDISP displays the header data of a LANDSAT image onto the terminal screen. This routine is generally used after GETHDR, get header, has been called.

All input values required to be displayed are stored in the arrays contained in the common block HEADER. The variable IFLAG determines whether these arrays are to be printed. If IFLAG is greater than or equal to 1, the arrays are not displayed.

FORMAT: CALL LMDISP(IFLAG)

LMDISP

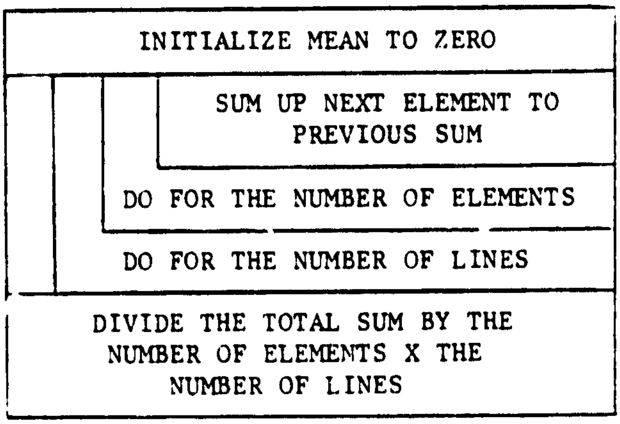
T	IS FLAG ≥ 1? F
NULL	DISPLAY SID, EXPODT, FORCEN, SUNANG, SUNANA, NADIR, HEADNG ON THE TERMINAL SCREEN

3.6.6 MEAN (Determine the Mean)

The subroutine MEAN calculates the mean of a group of array elements. The array IARRAY (Element, Line) contains the data values. IARRAY must be dimensioned as a 64 x 64 square 2-dimensional array. RWM is the calculated mean value. ISIZE is the number of elements and lines to be used. IES and ILS are the offsets into IARRAY from where the mean is to be taken.

FORMAT: CALL MEAN(IARRAY,ISIZE,RWM,ILS,IES)

MEAN



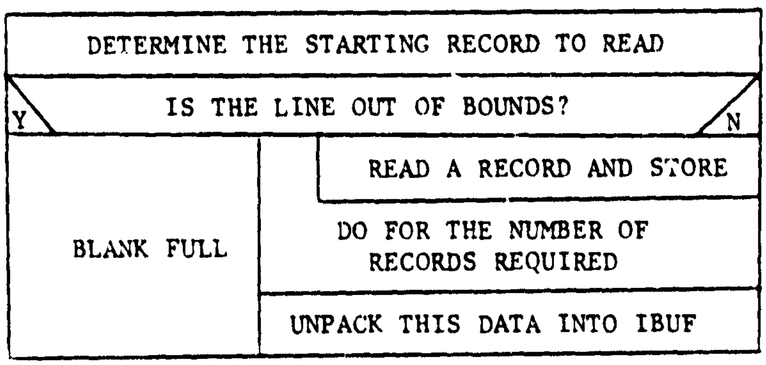
3.6.7 NXDREC (Next Disk Record)

The subroutine NSDREC is used to fill up an array that will contain image data. The image data will be read from unit 8. This subroutine is generally called by IMGET.

Each line of imagery is composed of seven 512-byte records. The variable IAVAR defines what the desired line is. NRED is the number of records needed to extract the desired data. NELEMS is the number of elements desired, and IES is the starting element on the line.

FORMAT: CALL NXDREC(IAVAR,NRED,NELEMS,IES)

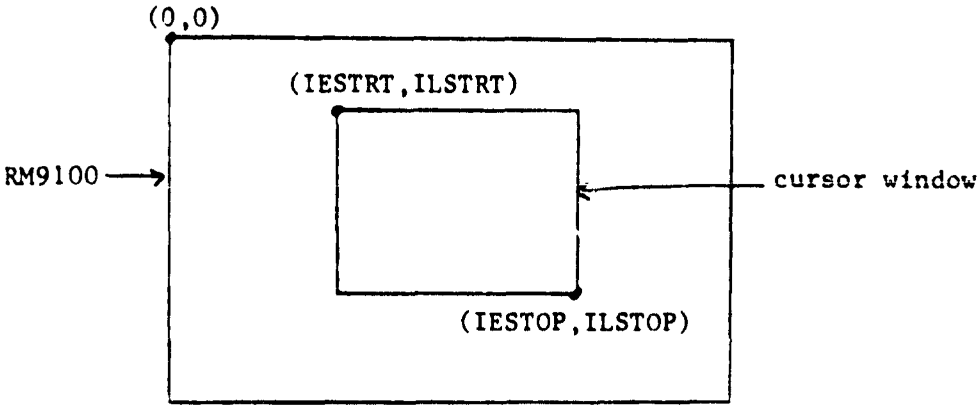
NXDREC



3.6.8 <u>POS</u> (Position Cursor)

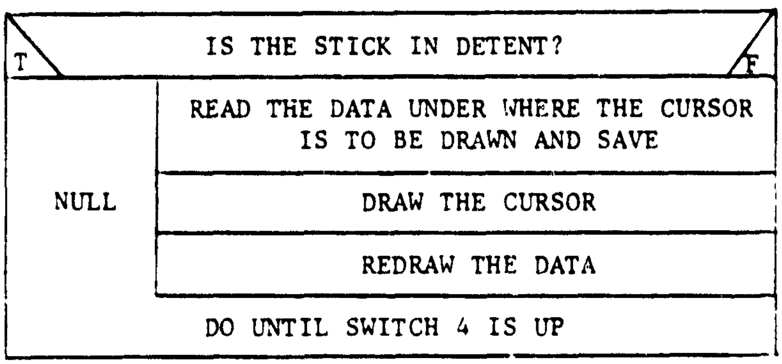
POS is used to allow the user to position the cursor on the RM9100 screen. This routine allows a 9 x 9 pixel cursor to be drawn and moved around by the user. When switch 4 is flipped up, then down, the user has positioned the cursor to the desired location, and control is returned to the calling routine.

IX and IY are the initial locations of the center of the cursor. IX and IY also receive the final values of the cursor center. ILSTRT, ILSTOP, IESTRT, and IESTOP define a rectangular area in which the cursor may move.



FORMAT: CALL POS(IX,IY,ILSTRT,ILSTOP,IESTRT,IESTOP)

<u>POS</u>



3.6.9 REDAT (Read in Data)

REDAT will read a M x N area from the RM9100 into a two-dimensional array IARRAY. The maximum size area is 64 x 64 pixels. The area is read beginning at line ILS and element IES. IBUF is a scratch array. IELEMS is the number of elements/line, and ILINES is the number of lines to read in.

FORMAT: CALL REDAT(IARRAY,IBUF,IELEMS,ILINES,IES,ILS)

REDAT

	READ A LINE OF DATA INTO THE SCRATCH ARRAY
	STORE THE SCRATCH ARRAY INTO THE FINAL BUFFER
DO FOR THE NUMBER OF LINES	

3.6.10 REDRAW (Redraw the Box)

REDRAW is used in conjuction with DRABOX. DRABOX draws a box around an area storing the data being overwritten by the box. REDRAW redraws this overwritten data.

ISIZE is the size of the box to be redrawn. IX and IY are the screen locations of the upper left hand corner of the area enclosed by the box. IBUF is the array containing the prestored data.

FORMAT: CALL REDRAW(ISIZE,IX,IY,IBUF)

REDRAW

DETERMINE BOX SIZE
REDRAW 4 SIDES OF THE BOX

3.6.11 SPCOEF/ SPLINE

The subroutine SPCOFF and the function SPLINE are designed to evaluate the natural cubic interpolatory spline fit to a set of data specified by the array of N nodes, XN, with corresponding function values in the array FN. The nodes of XN must be distinct. The spline is determined in SPCOEF and is evaluated in SPLINE. SPCOEF arranges the nodes in increasing order and stores this order in the array INDEX. Then XN(INDEX(1)) is the smallest node, XN(INDEX(2)) the next largest, etc. The array XN itself is not altered. SPCOEF then evalutates the array S of the N second derivatives needed to define the spline. The call arguments of the subroutine subprogram SPCOEF are:

SPCOEF(N,XN,FN,S,INDEX)

The remainder of the spline calculation is done by the function subprogram SPLINE, which has the call arguments:

SPLINE(N,XN,FN,S,INDEX)

Here N,XN,FN,S, and INDEX are as before; and X is the point at which we wish to evaluate the spline. Its value is returned by the function name SPLINE. The arrays XN, FN, and INDEX must be dimensioned in the calling program.

If X < XN(INDEX(1)), the function is approximated by a straight line which passes through the point (XN(INDEX(1)), FN(INDEX(1))) with a slope which agrees with the slope of the spline at that point. If X > XN(INDEX(N)), a straight line approximation is again used; this time it passes through (XN(INDEX(N)), FN(INDEX(N))) and has the slope of the spline at that point.

If we want to approximate a function $f(x)$ by a spline at several points x given a set of nodes XN, the routine SPCOEF need be called only once, and SPLINE is called once for each point x.

The SPCOEF/SPLINE program was defined in the book:

<u>NUMERICAL COMPUTING: An Introduction</u>, by Lawrence F. Shampine and Richard C. Allen, Jr., 1973, by W.B. Saunders.

3.7 I/O Routines

The I/O routines are those routines which are used strictly for I/O purposes. These routines are:

		SECTION
1.	ADJUST	3.7.2.1
2.	INIJOY	3.7.2.2
3.	READJY	3.7.2.3
4.	RESET	3.7.1.1
5.	RMCHCK	3.7.1.2
6.	RMERSE	3.7.1.3
7.	RMRDIM	3.7.1.4
8.	RMTEXT	3.7.1.5
9.	RMWRIM	3.7.1.6
10.	WIO	3.7.1.7

3.7.1 RAMTEK 9100 I/O ROUTINES

All RM9100 I/O routines use the WTQio and GETADR system I/O subroutines. The basic structure for a RM9100 I/O routine is the following:

(1) The command instruction is placed in a buffer.

(2) The address of the command buffer and the length of the buffer is stored in a parameter array. This is where GETADR is used.

(3) Depending on the function, the appropriate data from the parameter array is written/read to the RM9100. The operation of reading/writing is performed by using a WTQIO.

Two auxiliary routines that have been used by all RM9400 I/O subroutines are WIO and RMCHCK. These routines have been developed so that all I/O functions will utilize them and help conserve task space.

3.7.1.1 RESET (Reset the RM9100)

The RESET instruction is an RM9100 system clear function identical to the power-on (or hard reset) initialization sequence. All pending operations are discarded. The refresh memory will be erased.

FORMAT: CALL RESET

3.7 1.2 RMCHCK (Check RM9100 I/O)

The RMCHCK subroutine checks to see if the previous RM9400 I/O was correctly handled by the RM9100. The status words are contained in the common block IO. The directive status word, IDS, is equal to 1 if the directive was sent successfully. The I/O status word, IOST(1), is 1 if the issued I/O executed correctly. If either of these variables do not equal 1, the I/O was unsuccessful.

FORMAT: CALL RMCHCK

3.7.1.3 <u>RMERSE</u> (RM9100 Erase)

The RMERSE allows the programmer to erase a rectangular region on the RM9100. This region is defined by the variables ILSTRT, ILSTOP, IESTRT, and IESTOP. This is a software erase which uses the RMWRIM subroutine to write blank data into memory.

FORMAT: CALL RMERSE(ILSTRT,ILSTOP,IESTRT,IESTOP)

3.7.1.4 RMRDIM (RM9100 Read Image)

The RMRDIM subroutine allows the user to read a rectangular area from the RM9100 memory. Up to 2000 pixels may be read in any one call to RMRDIM.

ISCAN determines the primary and secondary scan directions.

ISCAN = 0 primary - left to right
secondary - top to bottom

ISCAN = 4 primary - top to bottom
secondary - left to right

IXS and IYS are the element and line values in memory at which the read is to begin. IDATA is the buffer in which data is to be filled, and ILEN is the number of pixels to be read.

FORMAT: CALL RMRDIM(ISCAN,IXS,IYS,IDATA,ILEN)

3.7.1.5 RMTEXT (Write Text on the RM9100)

The routine RMTEXT writes character text on the RM9100. In the CORALA program, the following variables should always have these values:

IADD = 0

IPOLAR = 0

ICHAN = -1

ISCALE = 0

ISCAN is the variable defining the scan direction in which the text will be written.

ISCAN = 0 primary - left to right
secondary - top to bottom

ISCAN = 4 primary - top to bottom
secondary - left to right

IXS and IYS are the upper left hand start point for the text. All text is written in 7 x 9 pixel fonts. The variable ICHAR is the address of the first character to be written. INCHAR is the length of the character string. INCHAR should always be an even number.

```
FORMAT:  CALL RMTEXT(IADD,IPOLAR,ICHAN,ISCAN,ISCALE,IXS,ICHAR,
                    INCHAR)
```

3.7.1.6 RMWRIM (RM9100 Write Image)

The RMWRIM writes to any rectangular area in the RM9400 in word mode. Up to 2000 pixels may be written in any one call to RMWRIM.

In the CORALA programs, ICHNL always equals -1, ISCALE always equals 0, and IFCTN always equals 0. ISCAN is the scan direction to be used for the write.

ISCAN = 0 primary - left to right
secondary - top to bottom

ISCAN = 4 primary - top to bottom
secondary - left to right

IXS and IYS are the element, line start where the write to memory is to occur. IDATA is the buffer address of the element(s) to be written, and ILEN is the number of pixels to be written.

```
FORMAT:  CALL RMWRIM(ICHNL,ISCAN,ISCALE,IFCTN,IXS,IYS,IDATA,ILEN)
```

3.7.1.7 <u>WIO</u> (Issue Wait for Queue I/O)

The WIO subroutine issues a PDP 11/70 WTQIO system directive. There are two arguments passed to this routine, IFCTN and IDEV. IFCTN specifies the I/O function to be executed, and IDEV is the device in which the I/O function is to occur.

This subroutine is used for all RM9400 I/O. The common block IO contains the other parameters required by the WTQIO instruction.

FORMAT: CALL WIO(IFCTN,IDEV)

FUNCTION CODES: "410 - Write Logical block (RM9400)

"1010 - Read Logical block (RM9400)

3.7.2 JOYSTICK I/O ROUTINES

The joystick device on the PDP 11/70 is terminal number TT4:. The joystick routines read this terminal and interpret the data. For data to be sent from TT4:, one of the switches marked 1 through 4 must be switched up.

3.7.2.1 ADJUST (Adjust Data)

The ADJUST subroutine is written in DEC MACRO assembly language. This routine adjusts the bits in the output data from the joystick. This routine is device dependent and called only by the subroutine READJY. The source code for ADJUST is found in the file ADJDAT.MAC.

3.7.2.2 <u>INIJOY</u> (Initialize the Joystick)

The INIJOY subroutine assigns a logical unit number to the joystick. This subroutine is called once in CORALA.

The parameter ILUN is the logical unit number to assign the joystick to. This unit number will be used by the routine READJY, to read the joystick device.

FORMAT: CALL INIJOY(ILUN)

3.7.2.3 <u>READJY</u> (Read the Joystick)

The READJY subroutine allows the user to read the joystick. Four words will be returned on each call. These words will be the switch word, X, Y, and Z words. The X, Y, and Z words have a range between 0 and 1023. When the stick is in detent, the X and Y words are 512. When the stick is up in the upper left corner, the X and Y words are 0. The Z word is set by the joy-stick knob. The switch word is bit oriented and follows the convention:

Bit Location	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Switch Location													X	X	X	X

SWITCH WORD

If bit 0 is set, switch 1 is up. If bit 1 is set, switch 2 is up, and so on. There are 4 switches.

The IWRDCT parameter must always be 4. ABUFF is the buffer to receive the 4 words of output. ILUN is the logical unit number of the joystick as initialized by INIJOY.

FORMAT: CALL READJY(ABUFF,IWRDCT,ILUN)

4.0 DATA BASE DEFINITION

A correlation analysis data base has been defined. This data base allows the user to store correlation run data. When the user specifies that the data base will be updated, certain parameters are selected for the user automatically. The test case number is automatically selected as well as the search area size of 64; landmark sizes of 48, 32, 24, 16, and 8; and the selection of subpixel registration on these landmarks. The test is run in band 1, .65 micron, and band 2, .85 micron.

The data base file is located in [112,5] CORRELATE.DAT. This is a direct access file, 3584 bytes per record. The first record is a header record. Element 1 of the header record contains the current number of tests that have been run. Subsequent records contain test data. Each test creates two records of data. One record for the test run in band 1 and one record for the test run in band 2.

The test data recorded is stored as integers or real values dependent upon their location in the record. The record definition is as follows:

CORRELATION ANALYSIS DATA BASE DEFINITION

Two records are written per test case. There is one header record located at record 1, records 2 and 3 contain data for test 1, records 4 and 5 contain data for test 2, and so on. . .

Correlation Analysis Data Base Definition (Continued)

Description	# of Words	LM Size	Start Buffer	Addr.
TEST CASE NUMBER	1		1	CORFIN
S.ID 1 2 CHARS/WORD	6		2	CORINI
S.ID 2 2 CHARS/WORD	6		8	
SCENE 1 START LINE	1		14	CORINI
SCENE 1 START ELEMENT	1		15	
SEARCH AREA 1 START LINE	1		16	CORINI
SEARCH AREA 1 START ELEM	1		17	
S.ID 1 LM % CLOUD *10	1	8	18	CORFIN
	1	16	19	
	1	24	20	
	1	32	21	
	1	48	22	
S.ID 2 S.A. % CLOUD*10	1		23	CORINI
S.ID 2 LM % CLOUD *10	1	8	24	CORFIN
	1	16	25	
	1	24	26	
	1	32	27	
	1	48	28	
CORRELATION TIME SSDA*10	1	8	29	CORFIN
	1	16	30	
	1	24	31	
	1	32	32	
	1	48	33	
CORRELATION TIME CLASS*10	1	8	34	CORFIN
	1	16	35	
	1	24	36	
	1	32	37	
	1	48	38	
SSDA ACT VS. AUTO CORR.	1	8	39	CORFIN
COEFFICIENTS*10000.	1	16	40	
	1	24	41	
	1	32	42	
	1	48	43	
CLASSICAL ACT. VS. AUTO	1	8	44	CORFIN
CORR. COEFFICIENTS*10000.	1	16	45	
	1	24	46	
	1	32	47	
	1	48	48	
L.M. START LINE S.ID 2	1		49	CORFIN

5.0 AUXILIARY PROGRAMS

Auxiliary programs were written to aid the user to control the flow of the CORALA program. These programs are:

(1) INICORALA - Initialize the CORALA data base

(2) AUTOCOR - This program creates the batch mode run stream for CORALA

(3) LANDSAT - This program reformats LANDSAT tapes to a deinterleaved format

(4) FULL - Displays a 1024 x 1024 section from a reformat LANDSAT image